

ADRIANO VITOR

**DETERMINAÇÃO DO ROTEAMENTO DOS ATENDIMENTOS DE UMA
EMPRESA DE COMERCIALIZAÇÃO AGRÍCOLA – VARIAÇÕES DE
SOLUÇÕES HEURÍSTICAS.**

CURITIBA

2007

ADRIANO VITOR

**DETERMINAÇÃO DO ROTEAMENTO DOS ATENDIMENTOS DE UMA
EMPRESA DE COMERCIALIZAÇÃO AGRÍCOLA – VARIAÇÕES DE
SOLUÇÕES HEURÍSTICAS.**

**Dissertação apresentada como requisito
parcial à obtenção do grau de Mestre em
Ciências, Curso de Pós-Graduação em
Métodos Numéricos em Engenharia,
Área de Concentração em Programação
Matemática, Setor de Ciências Exatas e
Setor de Tecnologia, Universidade
Federal do Paraná.**

**Orientador: Profº. Dr. Arinei Carlos
Lindbeck da Silva**

CURITIBA

2007

TERMO DE APROVAÇÃO

ADRIANO VITOR

“DETERMINAÇÃO DO ROTEAMENTO DOS ATENDIMENTOS DE UMA EMPRESA DE COMERCIALIZAÇÃO AGRÍCOLA – VARIAÇÕES DE SOLUÇÕES HEURÍSTICAS”.

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-Graduação em Métodos Numéricos em Engenharia – Área de Concentração em Programação Matemática, Setores de Tecnologia e de Ciências Exatas da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador:

Prof^o. Arinei Carlos Lindbeck da Silva, D.Eng.
Departamento de Matemática da UFPR

Prof^o. Luiz Fernando Nunes, Dr.
UTFPR / Professor Visitante

Prof. Paulo Henrique Siqueira, D.Eng.
Departamento de Desenho da UFPR

Curitiba, 17 de Setembro de 2007.

Dedico a todos os meus amigos e inimigos, que diretamente ou indiretamente me fortalecem a cada dia, proporcionando em boa quantidade, os motivos para eu querer viver.

AGRADECIMENTOS

Primeiramente à minha família, em especial, minha mãe querida pelo incentivo à busca por novos horizontes e ao meu avo materno que sempre foi meu espelho de vida.

Aos meus amigos: Gislaine, Solange, Juliano, Dirceu, Edivaldo, Amauri e tantos outros que durante o curso sempre estiveram ao meu lado. Tantas discussões produtivas e momentos de estudos que efetivamos juntos, sempre com o espírito de equipe, solidariedade e união que cultivamos desde os tempos de graduação.

Por fim a todos os professores que por meio da UFPR e da Fecilcam, proporcionaram a oportunidade para que eu pudesse realizar os créditos e a defesa deste trabalho.

Difícil encontrar o verdadeiro valor de um errante! Uns arriscam o ingênuo palpite de que está no coração; julgo eu ser o mais covarde dos órgãos, pois mais que depressa se desfaz do sangue a cada vez que lhe é confiado.

(o autor)

SUMÁRIO

LISTA DE TABELAS	ix
LISTA DE FIGURAS	x
RESUMO	xi
ABSTRACT	xii
1 INTRODUÇÃO	1
1.1 APRESENTAÇÃO DO PROBLEMA	2
1.1.1 Situação da Empresa	2
1.1.2 Os Clientes.....	3
1.1.3 Os Vendedores	4
1.1.4 As Rotas.....	4
1.1.5 As Distâncias	5
1.2 OBJETIVOS	7
1.2.1 Objetivo Geral.....	7
1.2.2 Objetivos Específicos	7
1.3 JUSTIFICATIVA	8
1.4 ESTRUTURA DO TRABALHO.....	8
2 REVISÃO DE LITERATURA.....	9
2.1 CONCEITOS BÁSICOS SOBRE GRAFOS	9
2.2 HISTÓRICO DA PROGRAMAÇÃO LINEAR.....	11
2.3 O PROBLEMA DO TRANSPORTE	12
2.4 PROBLEMA DO CAIXEIRO VIAJANTE (PCV)	14
2.5 MÉTODOS HEURÍSTICOS DE RESOLUÇÃO DO PCV	16
2.5.1 Algoritmo Heurístico que Utiliza os Savings de Clarke and Wright.....	17
2.5.2 Inserção mais Econômica	18
2.5.3 Melhoria 2 – <i>Opt</i> e 3 – <i>Opt</i>	19
2.5.4 Melhoria <i>k</i> – <i>Opt</i>	19
2.5.5 Procedimentos de Composição de Algoritmos.....	21
2.5.6 Heurística de Bellmore e Nemhauser	21
2.6 APLICAÇÃO DE REDES NEURAIS PARA RESOLUÇÃO DO PCV.....	22
2.7 O PROBLEMA DOS MÚLTIPLOS CAIXEIROS VIAJANTES (MPCV)	23
2.8 PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)	23
2.9 O ALGORITMO DE BUSCA DE CAMINHOS	25
2.10 O ALGORITMO DE FLOYD.....	26
2.11 ANÁLISE DE AGRUPAMENTOS (<i>CLUSTERS</i>).....	27
2.11.1 Medidas de Similaridade e Dissimilaridade.....	28
2.11.2 Método de Agrupamento	29
2.12 AVALIAÇÃO DA FORMAÇÃO DOS AGRUPAMENTOS.....	31
2.13 O ALGORITMO DAS P-MEDIANAS DE TEITZ E BART	31

2.14 O GPS.....	34
3 METODOLOGIA UTILIZADA PARA RESOLUÇÃO DO PROBLEMA REAL	36
3.1 OBTENÇÃO DA LOCALIZAÇÃO GEOGRÁFICA DOS CLIENTES	36
3.2 ELABORAÇÃO DA MATRIZ DE DISTÂNCIA ENTRE OS CLIENTES ...	39
3.2.1 Utilização do Algoritmo de Floyd.....	40
3.3 ELABORAÇÃO DAS ROTAS UTILIZANDO HEURÍSTICAS	41
3.3.1 A Heurística de Clarke and Wright	41
3.3.1.1 Elaboração da Restrição de União das Rotas para a Heurística.....	41
3.3.2 Heurística 2 – <i>Opt</i>	42
3.4 PROPOSTAS PARA UTILIZAÇÃO DO MODELO EXATO	42
3.4.1 Teitz e Bart, Primeiro Critério de Eliminação de Variáveis e o Modelo Exato	44
3.4.2 Teitz e Bart, Segundo Critério de Eliminação de Variáveis (Modelo PL → PLI) e o Modelo Exato	45
3.4.3 Técnica da Estatística (Análise de Clusters) e o Modelo Exato	46
4 IMPLEMENTAÇÃO COMPUTACIONAL	48
4.1 O MODELO DE CLUSTERIZAÇÃO PARA DISTRIBUIR OS 154 CLIENTES EM 5 GRUPOS DISTINTOS	48
4.2 O MODELO UTILIZADO PARA ELIMINAÇÃO DE VARIÁVEIS	49
4.3 O MODELO EXATO DE ROTEIRIZAÇÃO	51
5 ANÁLISES DOS RESULTADOS.....	53
5.1 RESULTADOS OBTIDOS UTILIZANDO HEURÍSTICAS	53
5.2 RESULTADOS OBTIDOS UTILIZANDO O MODELO EXATO	56
5.2.1 Resultados Obtidos Utilizando Teitz e Bart, Critério Superior e o Modelo Exato.....	56
5.2.2 Resultados Obtidos Utilizando Teitz e Bart, Modelo PL → PLI e o Modelo Exato.....	58
5.2.3 Resultados Obtidos Utilizando a Estatística, Modelo PL → PLI e o Modelo Exato.....	59
5.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS.....	61
6 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....	64
REFERÊNCIAS	65
LISTA DE ANEXOS	69

LISTA DE TABELAS

Tabela 1. 1 Coeficiente de Correção Associados às Áreas.....	6
Tabela 5. 1: Resultados Gerais de Acordo com a Heurística de Clarke and Wight	53
Tabela 5. 2: Conjunto de Rotas de Acordo com a Heurística de Clarke and Wright. ..	53
Tabela 5. 3: Resultados Gerais de Acordo Com a Heurística 2 – <i>Opt</i>	55
Tabela 5. 4: Conjunto de Rotas de Acordo Com a Heurística 2 – <i>Opt</i>	55
Tabela 5. 5: Resultados Gerais de Acordo Com Teitz e Bart e o Critério Superior.	57
Tabela 5. 6: Conjunto de Rotas de Acordo Com Teitz e Bart e o Critério Superior.	57
Tabela 5. 7: Resultados Gerais de Acordo Com Teitz e Bart e o Critério PL → PLI. ..	58
Tabela 5. 8: Conjunto de Rotas de Acordo Com Teitz e Bart e o Critério PL → PLI...	58
Tabela 5. 9: Resultados Gerais de Acordo Com a Estatística e o Critério PL → PLI. ..	60
Tabela 5. 10: Conjunto de Rotas de Acordo Com a Estatística e o Critério PL → PLI.	60
Tabela 5. 11: Resultados Gerais de Acordo Com a Formulação Exata Para Variáveis Não Inteiras.	62

LISTA DE FIGURAS

Figura 3. 1: Região de abrangência e Localização dos Clientes da Fertimourão	38
Figura 3. 2: Localização das cidades com sedes de clientes.....	39

RESUMO

Este trabalho propõe a resolução do problema real da determinação de rotas ótimas para o setor de vendas da Empresa Fertimourão Ltda, que atualmente realiza seus serviços de assistência técnica, venda de insumos agrícolas e compra de cereais de forma empírica. Para determinação das referidas rotas, realizou-se a coleta das informações digitalizadas (com o auxílio de um GPS), referente às estradas normalmente percorridas pelos vendedores da empresa, e as coordenadas geográficas das sedes ou residências dos clientes, em grande maioria, agricultores atuantes na região Centro Oeste do Estado do Paraná. Com base nos dados coletados, foi obtida a matriz de distâncias reais entre todos os pares de clientes para então ser realizada a roteirização, para a qual foram implementadas computacionalmente a Heurística de Clarke e Wright, a Heurística 2 – *Opt* e três propostas de variações no modelo exato de roteirização. Estas técnicas foram aplicadas sobre um conjunto de 153 clientes e mais a sede da empresa localizada na cidade de Campo Mourão/PR. As rotas foram planejadas para serem percorridas diariamente pelos 5 vendedores do setor de vendas, saindo toda manhã do pátio da sede retornando ao fim da tarde. Normalmente não se utiliza o modelo exato de roteirização para resolução de problemas relativamente grandes devido ao tempo computacional elevado, contudo neste trabalho aplicou-se três metodologias que possibilitaram a abordagem do problema em questão, desta forma, fornecendo resultados consideravelmente melhores que as heurísticas citadas acima.

Palavras-chave: Programação Matemática, Roteirização de Veículos, Distâncias Reais, Heurísticas.

ABSTRACT

This work aims the solution of a real problem that comes from the departament selling's itinerary of Fertimourão LTDA company. The company nowadays realizes works about technical assistance; sells agriculture poisons and buys cereals in an empirical form. To determine the itineraries, digitated information were collected by using the GPS. It was researched as a reference the way that the company sellers were used to take and the geography coordinates from the head office or the customers' house, in most case, agricultures from the west of Paraná state. According to the collected data, it was obtained the real distance between every pair of customers and then it was made an itinerary, to realize this it was used the heuristic method implemented by computer and also three proporsals on varieting the itineraries' model. These technics were applied with a group of 152 customers and also the head office located in Campo Mourão, Paraná. The five sellers were suposed to follow a planned itinerary every day, in the morning, leaving from the inner court of the head office and coming back in the afternoon. It is not usually used the exactly model of itinerary to solve the problems because of the computer process time. Although, in this work three methodologies were used to show the problems point. In this way, it was possible to achieve much better results comparing to the heuristic methods mentioned before.

Key-words: Mathematic Programation; vehicles Itinerary; Real Distances; Heuristic Methods.

1 INTRODUÇÃO

Este trabalho apresenta uma abordagem para o problema de roteamento de veículos aplicável a uma empresa de Campo Mourão, baseando-se nas técnicas de resolução deste tipo de problema conhecidas na literatura, para promover a otimização das rotas percorridas pelos funcionários do setor de venda, os quais trabalham atualmente sem auxílio de qualquer ferramenta de otimização e ou planejamento de ações.

A empresa de razão social Fertimourão Ltda., trabalha no ramo da agroindústria e desenvolve atividades de compra e venda de produtos agrícolas, e prestação de serviços. No caso das vendas, são em sua maioria insumos e defensivos agrícolas. Quanto às compras, estão ligadas aos cereais produzidos pelos clientes, enquanto que a prestação de serviço relaciona-se à armazenagem dos produtos.

Os agricultores, que são os clientes, estão localizados principalmente na região centro oeste do Estado do Paraná. Atualmente a empresa possui, além da sede em Campo Mourão, 09 entrepostos espalhados na região. Este trabalho tem como foco apenas a otimização das rotas dos vendedores da sede da empresa.

Para isso propõe-se à empresa uma metodologia de trabalho que difere das suas práticas atuais quanto à questão da forma com que são atribuídos os clientes aos vendedores. Atualmente as atribuições obedecem a critérios ligados à afinidade entre o cliente e o vendedor, e no modelo essas atribuições obedecem aos critérios de proximidade, tais critérios foram abordados por meio das metodologias descritas a seguir.

Na primeira abordagem, utilizou-se o algoritmo de Teitz e Bart para criação de pontos centrais e com base nestes pontos subdividiu-se o problema em grupos menores por meio de um algoritmo de designação, com posterior aplicação de um critério de eliminação de variáveis, diminuindo o tamanho dos problemas e permitindo a roteirização utilizando o modelo exato.

A segunda metodologia baseou-se em uma forma bastante semelhante à primeira, variando somente o critério de eliminação de variáveis que utilizou um algoritmo próprio denominado neste trabalho por critério “**PL**→**PLI**”.

A terceira abordagem deu-se pela aplicação das heurísticas dos *savings* de Clarke and Wright para construção das rotas e a utilização da heurística 2 – *Opt* para melhoria das rotas.

Por último foi proposta uma metodologia que utilizou a técnica estatística de Análise de Agrupamento (*Clusters*), o critério **PL**→**PLI** e a roteirização via modelo exato.

1.1 APRESENTAÇÃO DO PROBLEMA

1.1.1 Situação da Empresa

A Fertimourão encontra-se em constante expansão de suas atividades, tanto é que os gerentes e encarregados pelo setor de vendas da empresa perceberam que os trabalhos de atendimento aos agricultores, que sempre foram controlados de forma empírica, não se apresentavam tão satisfatórios, sendo passíveis de melhorias, pelo fato do volume de clientes e atendimentos tornarem-se cada vez maiores e as tomadas de decisões mais complexas.

Outra grande preocupação está no fato de que o crescimento da empresa obrigatoriamente impõe a necessidade de buscar novos clientes e o fortalecimento das relações comerciais com os atuais, para que o aumento das despesas possa ser contornado por um aumento equivalente nas fontes de renda. Este fato obriga a empresa a oferecer qualidade, que está ligada à eficiência no atendimento e entrega dos produtos comercializados, o que é feito em boa parte pelos próprios vendedores durante as visitas.

Diante de tudo isso a empresa sentiu-se fortemente impulsionada a procurar apoio na programação matemática para servir como ferramenta de auxílio na análise da eficiência das práticas atuais de trabalho e elaboração de estratégias de atuação para um futuro mais promissor.

1.1.2 Os Clientes

Os clientes são agricultores localizados no centro oeste do Estado do Paraná. Alguns deles são visitados mensalmente por serem assíduos compradores e vendedores, já outros, são ocasionais e recebem a visita dos vendedores somente algumas vezes por ano, devido à preferência do cliente por negociar com outra empresa do mesmo ramo, ou ainda à desatenção do vendedor. Há também o caso dos clientes que adquirem os produtos diretamente na empresa para uso em sua propriedade.

De forma geral, para que a empresa não corra o risco de perder clientes por falta de visita, a proposta deste trabalho é planejar um sistema de atendimento com rotas que prevêem uma visita mensal para todos os clientes independentemente de serem ocasionais ou não.

Destes clientes ocasionais, alguns estão na carteira¹ do vendedor, mas praticam tão poucas atividades comerciais com a empresa que não foram considerados neste trabalho. Assim, definiu-se para cada carteira cerca de 40 clientes, pois, foi a quantidade que englobou todos aqueles que praticam atividades comerciais consideráveis com a empresa.

¹ Nomenclatura dada ao conjunto de clientes associados a um vendedor

1.1.3 Os Vendedores

Para atender os clientes da sede, a empresa possui atualmente no seu quadro de funcionários, 8 vendedores com cada um atendendo uma carteira variável, com o mínimo de 40 e máximo de 100 clientes. A proposta é restringir este número a aproximadamente 40, pelos fatos expostos no item anterior.

Os vendedores se apresentam na empresa às 8 horas da manhã, tomam posse de seus veículos, visitam os clientes durante o dia, retornando no final da tarde deixando os mesmos no pátio para o dia seguinte. Portanto, as rotas são diárias e não duram mais que 6 horas (inclusive já considerando o tempo de atendimento).

Cada vendedor é livre para percorrer seus trajetos diários, porém, após a otimização ele passará a ter trajetos diários previamente estipulados, cabendo a ele somente optar pelo dia do mês a realizar cada trajeto.

Todos os vendedores recebem da empresa a mesma cota de combustível para realizarem seu trabalho, ressaltando que após aplicação do modelo esta situação pode mudar e a empresa poderá adaptar as cotas a cada caso.

1.1.4 As Rotas

Como há necessidade dos vendedores retornarem ao pátio da empresa no final de cada dia para devolver o veículo, as rotas recebem planejamento diário e o tempo máximo de duração de uma rota foi estipulado em 6 horas, pois, é a jornada de trabalho normal dos vendedores.

Antes da utilização do modelo, as rotas eram elaboradas toda manhã sem nenhum rigor matemático, levando em consideração a carteira de clientes e os pedidos de visitas.

Após a aplicação do modelo, são apresentadas ao vendedor as possíveis rotas a serem percorridas. Dessa forma, a cada manhã o vendedor pode optar por seguir uma

dessas rotas ou identificar entre elas, a que contenha um determinado cliente que tenha requisitado um atendimento para este dia.

A totalização das rotas associadas a um vendedor deve atender a cada cliente de sua carteira pelo menos uma vez por mês, assim há possibilidade de aumento nas vendas, pois, atualmente os vendedores não visitam os clientes com esta frequência.

É importante ressaltar que o modelo não leva em consideração a necessidade de um cliente ser visitado mais de uma vez por mês, ficando a critério do vendedor, criar uma alternativa para solucionar este problema.

1.1.5 As Distâncias

A matriz das distâncias entre todos os pontos (clientes e a Fertimourão) é o ponto de partida do trabalho e conseguir essas distâncias reais é praticamente impossível, sendo o maior problema enfrentado em trabalhos como este. Além disso, percorrer todos os caminhos possíveis entre os agricultores e a Fertimourão leva muito tempo e gera um custo muito alto.

Esta forma de abordar o problema seria a mais adequada, caso todos os caminhos existentes estivessem presentes no grafo que representa a malha viária da região de abrangência do problema, para então aplicar o algoritmo de Floyd² uma única vez ou o algoritmo de Dijkstra³ na quantidade de vezes necessárias para preencher toda a matriz de distâncias.

Contudo, na literatura costuma-se trabalhar com distâncias euclidianas, como no caso de Colombo (2001) que propôs uma solução para o problema de determinar a localização de centros de serviços emergenciais e realizar a subsequente roteirização dos veículos que realizam estes atendimentos. Foram tomadas as distâncias euclidianas com fator de correção para se obter as distâncias reais aproximadas.

² Algoritmo utilizado para determinar a menor distância entre todos os pares de vértices de um grafo. Este algoritmo será detalhado no capítulo II.

³ Algoritmo com finalidade parecida com a do Algoritmo de Floyd, a diferença é que aplicando o Floyd uma vez, depois de várias iterações tem-se a matriz de distâncias mínimas, já o Dijkstra deve ser aplicada várias vezes para se obter a mesma matriz.

Segundo Novaes (1993) é possível obter uma boa aproximação para a distância real com a aplicação de um fator de correção baseando na área onde estão localizadas as rotas. Em seu trabalho foram determinados coeficientes de correção r_E para rotas localizadas em áreas de 0.74, 0.98 e 18.44 km² de uma região urbana da cidade de São Paulo (Tabela 1.1).

Tabela 1. 1 Coeficiente de Correção Associados às Áreas.

Área da Região que contém a rota (km ²)	Coeficiente de correção (r_E)
0.74	1.5450
9.88	1.4510
18.44	1.3780

Fonte: Novaes, 1993

Os fatores de correção r_E permitem relacionar as distâncias euclidianas d_E com as distâncias reais aproximadas d_A , de modo que:

$$d_A \cong r_E \cdot d_E \quad (1.1)$$

No trabalho de Nunes (1998) foi utilizada a mesma metodologia que a emprega por Novaes (1993) e Colombo (2001) por se tratar de problemas semelhantes, onde a otimização ocorria em zonas urbanas, sendo que os resultados não foram comprometidos pela metodologia utilizada.

Como aqui, não havia a certeza de serem válidos estes fatores por se tratar de um caso de zonas rurais e os trabalhos referenciados terem sido desenvolvidos em zonas urbanas, a saída foi percorrer o máximo de caminhos possíveis para coletar dados das estradas e a localização geográfica dos clientes.

O trabalho de coleta de dados forneceu, no mínimo, um caminho real para cada par de clientes, e a partir disso, pôde-se trabalhar com a Matriz de Floyd considerando as distâncias reais entre todos os pares de clientes.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O presente trabalho tem como meta principal gerar rotas de atendimento para o setor de vendas da Fertimourão Ltda por meio das quatro metodologias propostas.

As metodologias levam em consideração todas as restrições pertinentes ao problema, buscando a redução dos gastos mensais da empresa com este setor e um aumento nos lucros, por oferecer a possibilidade de visitas a um maior número de clientes durante o mês.

1.2.2 Objetivos Específicos

- Elaborar metodologias que possam indicar melhores rotas a serem praticadas pela Fertimourão no seu setor de vendas, em relação a suas práticas atuais.
- Indicar para a empresa uma cota justa de combustível para cada vendedor, considerando a quilometragem percorrida e a quantidade de clientes atendidos por ele mensalmente;
- Fornecer parâmetros para que a empresa possa avaliar suas práticas atuais no setor em questão quanto a melhor forma de atuação;
- Contribuir com a redução dos custos de locomoção dos vendedores da empresa;
- Promover uma maior agilidade e abrangência no atendimento dos clientes;
- Servir como ferramenta para promover a organização e sistematização da metodologia de trabalho aplicada ao setor de vendas.

1.3 JUSTIFICATIVA

Considerando a expansão da empresa e a necessidade de sistematização de suas metodologias de trabalho para evitar um crescimento desordenado e, posteriormente, perder o controle da situação, a empresa necessita de uma ferramenta de auxílio, pois é praticamente impossível tomar decisões ótimas para problemas complexos de forma empírica. Percebe-se então, a importância de se aplicar as técnicas da programação matemática e estatística na definição das rotas da empresa.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos, sendo que o primeiro faz uma apresentação do problema relatando as condições gerais da empresa onde serão aplicados os conceitos abordados aqui, expõe também os objetivos almejados e as justificativas para a escolha do tema.

No capítulo II é feita uma revisão bibliográfica relacionada aos problemas de roteamento, expondo os principais métodos de resolução presentes na literatura.

No capítulo III são apresentados de forma mais detalhada os dados e a metodologia empregada para obtenção dos mesmos.

O capítulo IV apresenta a descrição e implementação computacional do modelo matemático e das heurísticas propostas para solucionar o problema das rotas do setor de vendas da Fertimourão Ltda.

No capítulo V é realizada a análise dos resultados obtidos e a comparação com a situação real atualmente praticada pela empresa.

Finalmente, no capítulo VI são feitas as conclusões e apresentadas sugestões para trabalhos futuros.

2 REVISÃO DE LITERATURA

Neste capítulo são abordadas algumas das técnicas atualmente utilizadas por outros autores, para resolução de problemas relacionados ao roteamento de veículos sendo que algumas destas técnicas fundamentam a metodologia aqui empregada.

2.1 CONCEITOS BÁSICOS SOBRE GRAFOS

Em Souza (1997) são apresentados os seguintes conceitos a respeito de Grafos, conceitos estes que poderão ser citados no decorrer deste trabalho:

- **Grafo ou Rede:**

É uma coleção de vértices (pontos ou nós) $x_1, x_2, x_3, \dots, x_n$ e uma coleção de linhas (arcos ou arestas) $a_1, a_2, a_3, \dots, a_m$ conectando todos ou alguns desses vértices.

O Grafo G é denotado e descrito por $G(X, A)$ onde X é o conjunto de vértices e A é o conjunto de linhas.

Quanto às linhas, o Grafo poderá ser direcionado, isto é, as linhas possuem uma direção, não direcionado quando suas linhas não possuem uma direção ou ainda misto.

- **Caminho Direcionado:**

Um caminho em um Grafo direcionado é qualquer seqüência de arcos onde o vértice final de um é o vértice inicial do seguinte.

- **Caminho Simples:**

É um caminho no qual um arco não figura mais que uma vez

- **Caminho Elementar:**

É um caminho no qual um vértice não figura mais que uma vez.

- **Cadeia não direcionada:**

Uma cadeia é um Grafo em qualquer sequência de arestas na qual cada aresta é conectada com suas arestas adjacentes através dos seus dois vértices terminais.

- **Grafo Conexo:**

Um Grafo é dito conexo se existir entre quaisquer pares de nós do Grafo G pelo menos uma cadeia. Caso contrário o Grafo é dito não conexo.

- **Custo:**

Se a cada arco (x_i, x_j) de um Grafo for associado um custo $c_{i,j}$, o custo de um caminho μ é igual à soma dos custos dos arcos que formam este caminho.

- **Circuito:**

É um caminho no qual o vértice inicial coincide com o vértice final.

- **Circuito Elementar:**

É um circuito no qual cada vértice só aparece uma vez, exceto o vértice inicial e o final, que no caso, são coincidentes.

- **Circuito Hamiltoniano:**

É um circuito elementar que contém todos os n vértices do Grafo G .

- **Ciclo:**

É uma parte não direcionada de um circuito. Assim, um ciclo é uma sequência de arcos onde o vértice inicial e final, são os mesmos.

- **Matriz de Adjacência:**

Para um Grafo $G(X, A)$, sua matriz de adjacência, denotada por $A = [a_{ij}]$, é definida como:

$$a_{ij} = \begin{cases} 1 & \text{se o arco } (x_i, x_j) \text{ existir em } G \\ 0 & \text{se o arco } (x_i, x_j) \text{ não existir em } G \end{cases} \quad (2.1)$$

- **Matriz de Incidência:**

Para um Grafo $G(X, A)$, sua matriz de incidência, denotada por $B = [b_{ij}]$, é uma matriz $n \times m$, onde m é o número de arcos, definida como:

$$b_{ij} = \begin{cases} 1 & \text{se } x_i \text{ é o vértice inicial de } a_j \\ -1 & \text{se } x_i \text{ é o vértice final de } a_j \\ 0 & \text{se } x_i \text{ não é um vértice terminal de } a_j, \text{ ou se } a \text{ é um laço} \end{cases} \quad (2.2)$$

Se $G(X, A)$ for um Grafo não direcionado, a matriz de incidência é definida como acima, exceto trocando “-1” por “1”.

2.2 HISTÓRICO DA PROGRAMAÇÃO LINEAR

A Programação Linear surgiu como um dos mais importantes ramos da Programação Matemática com uma vasta aplicação prática. Inovações da última metade do século passado tornaram os algoritmos de Programação Linear, eficientes e favoráveis para a resolução de uma larga variedade de problemas envolvendo questões de decisão em vários domínios.

O problema de otimizar uma função linear sujeita às restrições lineares teve a sua origem com os estudos de Fourier sobre sistemas lineares de inequações em 1826. No entanto, só em 1939 Kantorovich fez notar a importância prática destes problemas, criando um algoritmo para a sua solução.

Num documento cujo objetivo era expor conceitos, Kantorovich apresentou exemplos para a aplicação da Programação Linear. Um desses exemplos envolvia a distribuição de fluxos de carga (distribuídos através de veículos de transporte), usando diferentes rotas em redes rodoviárias de forma a satisfazer os requisitos e as restrições de capacidade das rotas, minimizando o consumo de combustível.

Esse tipo de problema teve o seu auge com George Dantzig na década de 1940, consultor de matemática do *US Air Force Controller*, e com o prêmio Nobel da Economia George Stigler, que formulou o problema das dietas como um problema de mistura de componentes.

Dantzig não só formulou o problema de Programação Linear, mas também criou o Algoritmo Simplex para a sua solução em 1947.

Ainda em 1947, Koopmans mostrou que a Programação Linear é um modelo apropriado para a análise da teoria econômica clássica. Entretanto, nos EUA, Frank L. Hitchcock apresentou o que é hoje a formulação base do problema de transporte. Independentemente, o professor Koopmans formulou o mesmo problema em ligação com o seu trabalho efetuado na “*Combined Shipping Adjustment Board*”. Por isso, o problema de transporte é referido, na literatura científica, quer como problema de transporte de Hitchcock, quer como problema de transporte de Hitchcock-Koopmans.

Em 1956, Alex Orden propôs a generalização do modelo de transporte em que eram permitidos pontos de transbordo de carga. Esta formulação é conhecida hoje como um problema de transbordo sem limite de capacidade.

2.3 O PROBLEMA DO TRANSPORTE

Os problemas de transporte, normalmente caracterizam-se pela existência de um conjunto de pontos ou clientes que devem ser visitados por um veículo que normalmente sai de um ponto origem e retorna a este mesmo ponto tendo visitado todos os outros uma só vez e percorrido a menor distância possível.

Este problema também é conhecido como problema do caixeiro viajante e muitas vezes se torna impossível a sua resolução por um método exato quando a quantidade de pontos visitados é muito grande. No entanto existem alguns algoritmos capazes de fornecer soluções para grandes instâncias do problema.

Segundo Goldbarg e Luna (2000), o problema do caixeiro viajante pode ser aplicado em inúmeros problemas práticos, como por exemplo:

- Programação de operações de máquinas em manufatura;
- Programação de transporte entre células de manufatura;
- Otimização do movimento de ferramentas de corte;

- Otimização de perfurações e soldas de circuitos impressos;
- Na maioria dos problemas de roteamento de veículos;
- Na solução de problemas de sequenciamento de DNA (Genoma);
- Na solução de problemas de programação e distribuição de tarefas em plantas;
- Trabalhos administrativos.

Já para problemas de roteamento, existe um grande número de aplicações práticas presentes em diversos setores, como por exemplo:

- Distribuição de jornais, manufaturados, produtos diversos, bebidas, valores, produtos químicos, pão, gás, derivados de petróleo, alimentos, material fotográfico, vagões ferroviários, leite, concreto, etc.
- Transporte de pedras, escolar, coletivo (táxi, ônibus, trens de metrô), bens pessoais (mudanças), etc.
- Recolhimento de lixo, borracha, garrafas de leite, cana-de-açúcar, etc.
- Entregas de correspondências, pizzas, congelados, correspondências bancárias, etc.
- Serviços de emergência, patrulhamento policial, proteção contra incêndio, medidores elétricos, limpeza de rua com veículos vassoura, limpeza de gelo em ruas e estradas, etc.
- Roteamento de helicópteros, linhas aéreas, navios de curso longo (petroleiros) e cabotagem, auditores bancários, entre células de manufatura flexível, de robôs em manufatura, de pacotes em redes de computadores, de fluxo de comunicações em redes de telecomunicações, etc.
- Projeto de redes de telecomunicações em anéis.

2.4 PROBLEMA DO CAIXEIRO VIAJANTE (PCV)

Segundo Souza (1997) não se sabe quem foi o precursor do termo “Caixeiro Viajante”, mas sua divulgação se deve principalmente a Merrill Flood que o fez motivado em propor um desafio intelectual por modelos matemáticos como o da Teoria dos Jogos. Outro motivo pelo qual o PCV popularizou-se bastante está ligado aos problemas de designação. O primeiro artigo que propôs a solução em larga escala do problema do Caixeiro Viajante foi publicado por Dantzig, Fulkerson e Johnson em 1954 no Jornal da Sociedade de Pesquisa Operacional da América, que foi o principal evento da história da Otimização Combinatória.

O problema do caixeiro viajante é um dos problemas mais estudados em Otimização Combinatória. O objetivo do PCV é encontrar em um Grafo $G(V,A)$ um circuito hamiltoniano de menor custo. Encontra-se abaixo uma formulação deste problema elaborada por Golden (1977):

Assumindo que os custos são simétricos, isto é, $c_{ij} = c_{ji}$ e estabelecendo que $c_{ii} = \infty$ para $i = 1, 2, 3, \dots, n$, o problema consiste em formar uma rota passando por todos os nós, começando e terminando na origem, nó 1, que forneça a distância ou custo total mínimo. Sejam n o número de nós da rede, a_i a quantidade de arcos que possuem o nó i como vértice inicial, b_j a quantidade de arcos que possuem o nó j como vértice final, e x_{ij} a variável de decisão do problema, estabelecida como segue:

$$x_{ij} = \begin{cases} 1 & \text{se o arco } ij \text{ está na rota} \\ 0 & \text{caso contrário} \end{cases} \quad (2.3)$$

$$\begin{aligned}
& \text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
& \text{s.a :} \\
& \sum_{i=1}^n x_{ij} = b_j \quad (j = 1, 2, \dots, n) \\
& \sum_{j=1}^n x_{ij} = a_i \quad (i = 1, 2, \dots, n) \\
& X = [x_{ij}] \in S \\
& x_{ij} \in \{0,1\} \quad (i, j = 1, 2, \dots, n)
\end{aligned}$$

A formulação do problema seleciona uma matriz $X = [x_{ij}]$ de variáveis de decisão tal que exatamente um arco (ij) emana de cada nó i e exatamente um arco (ij) é direcionado para cada nó j . Isso implica numa designação de cada nó para seu nó sucessor na rota que é estabelecido pelo primeiro e segundo conjunto de restrições. O quarto conjunto de restrições define o problema de Programação Linear inteiro.

Os requisitos para designação, entretanto, não garantem que a matriz X corresponda a um circuito hamiltoniano. Para eliminar a possibilidade de formação de circuitos não hamiltoniano, restrições adicionais fornecidas pelo terceiro conjunto de restrições, são impostas nas escolhas para seleção dos arcos na matriz X .

O conjunto S pode ser formado por quaisquer restrições que proíbam a formação de circuitos não hamiltoniano que satisfaçam aos segundo, terceiro e quinto conjuntos de restrições. Tais restrições são chamadas de restrições de quebra de sub-rota.

Há uma segunda formulação presente nas obras de Christofides (1975) e Christofides (1979) apud Colombo (2001) e Souza (1997) onde o modelo fica da seguinte forma:

$$\begin{aligned}
& \text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
& \text{s.a :} \\
& \sum_{j=1}^n x_{1j} = 2 \\
& \sum_{j=1}^n x_{ij} + \sum_{j=1}^n x_{ki} = 2 \quad (i, k = 1, 2, \dots, n) \\
& \sum_{i=1}^n \sum_{j=1}^n x_{ij} = n \\
& \sum_{i=1}^n x_{ij} = 1 \quad (j = 1, 2, \dots, n) \\
& \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, 2, \dots, n) \\
& X = [x_{ij}] \in S \\
& x_{ij} \in \{0,1\} \quad (i, j = 1, 2, \dots, n)
\end{aligned} \tag{2.4}$$

2.5 MÉTODOS HEURÍSTICOS DE RESOLUÇÃO DO PCV

Quando se pretende resolver um PCV relativamente grande, se torna inviável utilizar métodos exatos, pois o tempo computacional para a obtenção da resposta ótima é muito elevado. Nestes casos convém utilizar heurísticas, porque além de fornecerem boas soluções são mais rápidas e não possuem restrições quanto ao tamanho do problema.

Souza (1997) cita que existem três grandes classes de métodos heurísticos mais conhecidos na literatura para a resolução do PCV:

- Procedimentos de construção de rotas ótimas ou quase ótimas considerando a matriz de distâncias (custos);
- Procedimentos de melhorias de rotas já existentes;

- Procedimentos de composição de algoritmos, que constroem uma rota inicial com o auxílio de um dos procedimentos de construção de rotas e utilizam melhorias para obter um resultado mais eficiente, evitando cruzamento entre rotas e outros problemas.

A seguir serão detalhados alguns desses métodos heurísticos para a resolução do PCV citados por Souza (1997).

2.5.1 Algoritmo Heurístico que Utiliza os Savings de Clarke and Wright

O algoritmo dos *savings* de Clarke and Wright é um procedimento que constrói rotas simultaneamente, e a cada iteração vai efetuando trocas nestas rotas por um conjunto melhor de rotas.

Inicia-se o processo com um veículo para atender cada nó individualmente. Verifica-se então a possibilidade de trocar dois veículos que atendem dois nós por apenas um veículo.

No próximo passo, as rotas vão sendo construídas de acordo com o maior saving $s(i, j)$, calculado da seguinte forma:

$$s(i, j) = [2c(0, i) + 2c(0, j)] - [c(0, i) + 2c(0, j) + c(i, j)] = c(0, i) + c(0, j) - c(i, j) \quad (2.5)$$

ALGORITMO:

Passo 1: Selecione qualquer nó como depósito central (nó 0), inicialize as n rotas, ligando cada nó, exclusivamente ao depósito obtendo-se as rotas $(0 - i - 0)$, para $i = 1, \dots, n$.

Passo 2: Calcule os *savings* entre os pontos de demanda P_i e P_j , da seguinte forma:

$$s(i, j) = c(0, i) + c(0, j) - c(i, j) \quad \forall i, j \quad (2.6)$$

Passo 3: Obtenha os *savings* do maior para o menor;

Passo 4: Começando do topo da lista dos *savings* e movendo para baixo, forme sub-rotas considerando que:

- P_i e P_j sejam nós “de ponta” de suas respectivas sub-rotas;
- P_i e P_j não devem estar alocados à mesma sub-rota;
- $d(R_i) + d(R_j) < K_V$, ou seja, a demanda do veículo da sub-rota i unida à demanda do veículo da sub-rota j não deve exceder à capacidade de cada veículo;

Passo 5: Se as três condições do passo anterior forem satisfeitas, unir as sub-rotas dos pontos P_i e P_j ;

Passo 6: Se não existir $s(i, j)$ que satisfaça as condições do passo 4, pare a execução do algoritmo. Caso contrário volte ao passo 4.

Segundo Goldbarg e Luna (2000), a aplicação mais eficiente dessa heurística para o Problema do Caixeiro Viajante sugere a necessidade de um grafo completo, onde cada vértice possui ligação com os demais vértices no grafo, sendo, portanto uma abordagem razoável para o PCV simétrico e euclidiano, que se aplica a este trabalho.

2.5.2 Inserção mais Econômica

Passo 1: Comece com uma sub-rota consistindo de um nó somente;

Passo 2: encontre o nó k tal que c_{ik} é mínimo e forneça a rota $(i - k - i)$;

Passo 3 (Passo da inserção): Encontre o arco (i, j) na sub-rota que minimize $c_{ik} + c_{kj} - c_{ij}$, então insira k entre i e j .

Passo 4: Vá para o passo 3 até que se tenha um circuito hamiltoniano.

2.5.3 Melhoria 2-Opt e 3-Opt

Os procedimentos de melhoria de rotas mais utilizados são o 2-Opt e 3-Opt que foram introduzidos por Lin (1965) e o procedimento k -Opt, apresentado posteriormente por Lin e Kernighan (1973).

ALGORITMO:

Passo 1: Encontre uma solução inicial por qualquer um dos métodos de construção de rotas;

Passo 2: Faça a melhoria das rotas utilizando uma das heurísticas de trocas;

Passo 3: Continue com o passo 2 até que não haja mais trocas a serem efetuadas. O procedimento de trocas é o seguinte: Para um dado k , definimos uma k troca de rota consistindo da deleção de k arcos numa rota que serão trocados por k outros arcos de modo a formar uma nova rota. Uma rota é k -ótima se não for mais possível efetuar trocas para melhorar a distância total viajada. Geralmente o 3-Opt chega bem próximo da solução e o 4-Opt gera uma melhoria que não justifica o custo computacional.

2.5.4 Melhoria k -Opt

Este algoritmo é mais eficiente que os outros propostos no tópico anterior, pois a cada iteração ele decide qual será o valor de k ótimo para efetuar as trocas entre os arcos.

ALGORITMO:

Seja G_P^* a melhoria ou segurança que pode ser realizada, se os P arcos atuais podem ser trocados de forma a gerar uma nova rota melhor que a anterior:

Passo 1: Escolha um roteiro inicial

Passo 2: Faça $G_0^* = 0$. Selecione qualquer nó como nó inicial (l) e considere qualquer um dos arcos adjacentes a este nó, da solução atual (l, k) para remover. Faça $P=1$.

Passo 3: Dos arcos restantes, escolha um arco que não esteja na solução atual (k, i) , tal que $g_1 = c_{lk} - c_{ki} > 0$. O arco é escolhido de modo a maximizar g_1 .

Passo 4: Tendo escolhido (l, k) para sair e (k, l) para entrar na solução da iteração anterior, o arco que deixará o circuito da P -ésima iteração é retirado. Adicionando o arco (j, l) , reconstrói-se a rota. $G_1^* = g_1 + c_{ij} - c_{jl}$ se $g_1 > 0$ e $G_1^* > 0$, faça $P = P + 1$.

Passo 5: O arco (j, l) não é necessariamente o arco escolhido para entrar na solução desta iteração. Procura-se o nó q que maximize $g_q = c_{ij} - c_{iq}$. Se $q = m$, o arco escolhido para entrar é (j, m) . Calcule $G_P = \sum_{s=1}^P g_s$ e então, $G_P^* = G_P + c_{mn} - c_{nl}$.

Seja $G_P^* = \max\{G_0^*, G_1^*, G_2^* \dots G_P^*\}$. Faça $P = P + 1$ e repita os passos 5 até que:

- Não existam mais trocas possíveis que melhorem as rotas.
- A solução corrente já é uma rota ótima.
- $G_P \leq 0$ ou
- $G_P \leq G^*$

Se uma destas condições estiver satisfeitas, constrói-se a rota associada com a melhor solução se $\{G_0^*, G_1^*, G_2^* \dots G_P^*\}$.

O procedimento é repetido até que todos os nós tenham sido utilizados como nó inicial e não existam mais melhorias a serem efetuadas na rota.

2.5.5 Procedimentos de Composição de Algoritmos

Segundo Souza (1997), o procedimento de composição de algoritmos é relativamente rápido computacionalmente e fornece excelentes resultados. A composição de algoritmos consiste do seguinte:

Passo 1: Obter uma solução inicial usando algum dos procedimentos de construção de rotas;

Passo 2: Aplicar o 2 – Opt para a rota encontrada no passo 1;

Passo 3: Aplicar o 3 – Opt para a rota encontrada no passo 2;

A principal função da composição de procedimentos é melhorar rapidamente a solução inicial e esperar que o 2 – Opt e 3 – Opt forneçam uma solução quase ótima. O algoritmo 3 – Opt pode ser utilizado somente uma vez, ou até que não haja mais trocas a serem efetuadas de forma a melhorar a solução obtida.

2.5.6 Heurística de Bellmore e Nemhauser

Além das heurísticas já citadas, Ganhoto (2004) apresenta também a Heurística de Bellmore e Nemhauser, publicada em 1968, onde a chamam de heurística construtiva que elege um vértice inicial de maneira arbitrária e na sequência inclui outros vértices no circuito em formação, seguindo o critério do vértice ou vizinho mais próximo.

Goldbarg e Luna (2000) mencionam que existe uma variação para minimizar o efeito da influência da escolha do vértice inicial, mas apesar de variações podemos escrever o algoritmo da seguinte forma:

ALGORITMO:

Passo 1: Escolha um vértice inicial, encontre o vértice mais próximo dele e inicie a construção do circuito através da ligação desses dois extremos.

Passo 2: Encontre o vértice mais próximo de um dos vértices extremos incluídos na solução; insira este vértice na rota em construção ligando-o com seu vizinho mais próximo.

Passo 3: Se todos os vértices foram incluídos, complete com a aresta que liga os vértices extremos e pare.

Passo 4: Caso contrário retorne ao passo 2.

2.6 APLICAÇÃO DE REDES NEURAIIS PARA RESOLUÇÃO DO PCV

Siqueira (2005) define como sendo uma Rede Neural Artificial (RNA) o conjunto de sistema de processamento de informações com características de aprendizado semelhantes às Redes Neurais Biológicas. Geralmente, os modelos RNA's possuem inspiração biológica, mas são algoritmos computacionais que representam o aprendizado humano de maneira bastante simples, por meio dos elementos de processamento chamados de neurônios.

Ainda de acordo Siqueira (2005) existe diversos tipos de Redes Neurais, sendo que a Rede Recorrente de *Wang* e a Rede de *Kohonen* foram as redes utilizadas em seu trabalho porque, segundo ele, a implementação computacional dessas redes não é complexa e os resultados de aplicações mostram-se técnicas eficientes para o resolução de problemas do gênero.

Mulder e Wunsch II (2003) apresentam a combinação de Rede Neural do tipo “dividir e conquistar”, com a técnica de melhoria proposta por Lin e Kernighan (1973) para resolver o Problema do Caixeiro Viajante simétrico. Neste trabalho os autores fazem comparações desta técnica com técnicas semelhantes encontradas na literatura.

2.7 O PROBLEMA DOS MÚLTIPLOS CAIXEIROS VIAJANTES (MPCV)

O MPCV é uma generalização do PCV e está mais próximo de uma situação real em que normalmente se tem uma frota de M veículos e deseja-se visitar todos os n nós de um grafo, tal que a distância total viajada pelos M veículos seja mínima. Cada caixeiro deve viajar uma sub-rota de nós, os quais incluem um depósito comum, e cada nó, exceto o primeiro deve ser visitado por exatamente um caixeiro.

A formulação exata para o MPCV segundo Bodin (1983) é uma extensão natural do PCV e pode ser expressa por:

$$\begin{aligned}
 &\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
 &\text{s.a.:} \\
 &\sum_{i=1}^n x_{ij} = b_j \begin{cases} M & \text{se } j = 1 \\ 1 & \text{se } j = 2, 3, \dots, n \end{cases} \\
 &\sum_{j=1}^n x_{ij} = a_i \begin{cases} M & \text{se } i = 1 \\ 1 & \text{se } i = 2, 3, \dots, n \end{cases} \\
 &X = [x_{ij}] \in S \\
 &x_{ij} = \{0,1\} \quad \forall i, j = 1, 2, 3, \dots, n
 \end{aligned} \tag{2.7}$$

O conjunto S de restrições, que quebram as sub-rotas (dentro de uma mesma rota), pode ser o mesmo utilizado para a solução do PCV.

2.8 PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

Ganhoto (2004) relata que o Problema de Roteamento de Veículos é um dos problemas de Otimização Combinatória que desperta grande interesse devido sua aplicação prática.

O PRV assemelha-se muito ao PCV e ao MPCV, todos procuram a melhor maneira de atender a um conjunto de clientes de forma que a distância total percorrida seja a menor possível.

No caso do PCV, não existem restrições de tempo, de capacidade ou demanda e dispõe-se somente de um veículo, já o MPCV utiliza-se de vários veículos para realizar as visitas havendo restrições de demanda para os clientes. Por fim o PRV é o mais abrangente dos três e considera além dos muitos veículos, restrições de tempo para as rotas e de capacidade para os veículos e demanda para os clientes.

Desta forma Ganhoto (2004) afirma que devido a suas características, o PRV também é um problema de difícil solução computacional e que os algoritmos existentes para sua resolução de forma ótima conseguem resolver instâncias ainda menores que os algoritmos que resolvem de forma exata o PCV e MPCV.

Neste cenário os vértices do grafo podem assumir o papel de depósito ou de postos de coleta/entrega de produtos. As arestas do grafo representam as ruas ou estradas que conectam os vértices fornecendo as distâncias entre os mesmos.

As demandas que podem estar tanto nos vértices quanto nas arestas, devem ser supridas por veículos que possuem capacidade própria.

O tempo de entrega, em alguns casos também é estipulado respeitando-se ainda o tempo total de viagem e o tempo para eventuais pausas como no caso do problema abordado neste trabalho.

De acordo com Chong (2001), o PRV foi inicialmente apresentado por Garvin et al (1957), para auxiliar no processo de distribuição de gasolina para postos de serviços, utilizando uma frota de veículos com capacidades distintas. Na ocasião, Garvin et al. (1957) utilizaram o modelo de Programação Linear para a resolução de uma variedade de problemas nas áreas de perfuração e produção manufatura, marketing e distribuição da indústria petroquímica.

Beltrami e Bodin (1974) também relatam uma das primeiras aplicações do PRV na coleta de lixo de Nova Iorque.

Russel e Igo (1979) abordaram problemas de coleta de lixo. Em ambos os trabalhos, as heurísticas propostas são baseadas no algoritmo de Clarke e Wright (1964) e movimentos $k - Opt$ de Lin e Kernighan (1973).

Ganhoto (2004) ressalta que apesar de existirem diversas variações para o PRV, umas com múltiplos depósitos, outras com restrições de tempo, algumas ainda utilizando frotas de veículos com capacidades diferentes e outras em que as demandas se encontram nas arestas. Neste trabalho nos concentraremos no estudo do PRV com um único depósito, demanda nos vértices e frotas não-homogêneas (veículos com capacidades diferenciadas).

2.9 O ALGORITMO DE BUSCA DE CAMINHOS

Na teoria de grafos, o problema do caminho mínimo consiste na minimização do custo de travessia de um grafo entre dois nós (ou vértices); custo este dado pela soma dos pesos de cada aresta percorrida.

Os algoritmos especializados em solucionar o problema do caminho mínimo são eventualmente chamados de algoritmos de busca de caminhos. Entre os algoritmos dessa classe, os mais conhecidos são:

1. **Algoritmo de Dijkstra:** Resolve o problema com um vértice-fonte em grafos cujas arestas tenham peso maior ou igual a zero. Sem reduzir o desempenho, este algoritmo é capaz de determinar o caminho mínimo, partindo de um vértice de início v para todos os outros vértices do grafo.
2. **Algoritmo de Bellman-Ford:** Resolve o problema para grafos com um vértice-fonte e arestas que podem ter pesos negativos.
3. **Algoritmo A*:** um algoritmo heurístico que calcula o caminho mínimo com um vértice-fonte.
4. **Algoritmo de Floyd-Warshall:** Determina a distância entre todos os pares de vértices de um grafo.
5. **Algoritmo de Johnson:** Determina a distância entre todos os pares de vértices de um grafo, pode ser mais eficiente que o algoritmo de Floyd-Warshall em grafos esparsos.

De acordo com Costa (1997) o algoritmo de Dijkstra é um dos mais eficientes para se determinar o caminho mínimo entre dois nós de um grafo.

2.10 O ALGORITMO DE FLOYD

Este procedimento, devido a Floyd (1962), baseia-se numa seqüência de n transformações (iterações) sobre uma matriz inicial de custos $C=[c_{ij}]$, de dimensão $n \times n$, até que na k -ésima iteração, a matriz representa o caminho de distância mínima entre todos os pares de vértices com a restrição de que o caminho entre x_i e x_j (para quaisquer x_i e x_j) contém apenas nós do conjunto restrito $\{x_1, x_2, \dots, x_k\}$ como intermediários.

ALGORITMO:

Passo 1: Inicializar a matriz de custos $C=[c_{ij}]$ da seguinte maneira:

$$c_{ij} = \begin{cases} 0, & \forall (x_i, x_i) \\ \infty, & \text{se } (x_i, x_j) \\ \text{custo associado ao arco } (x_i, x_j) \end{cases} \quad (2.8)$$

Passo 2: Fazer $k = 0$

Passo 3: $k = k + 1$

Passo 4: Para todo $i \neq k$ tal que $c_{ik} \neq \infty$ e todo $j \neq k$ tal que $c_{kj} \neq \infty$ realizar a operação $c_{ij} = \min\{c_{ij}, (c_{ik} + c_{kj})\}$

Passo 5.a: Se $k = n$, parar. A solução alcançada e $[c_{ij}]$ fornece os custos mínimos para cada par de vértices.

Passo 5.b: Se $k < n$, voltar ao Passo 2.

Segundo Costa (1997), este algoritmo requer um tempo computacional proporcional a n^3 , sendo n o número de vértices do grafo em questão.

Os trajetos mínimos entre dois nós x_i e x_j , podem ser obtidos dos caminhos de mínimo custo usando uma relação recursiva similar a equação dada pelo passo 4.

Hu (1969) fornece um procedimento usado para armazenar informações sobre os trajetos envolvendo o armazenamento e atualização de uma segunda matriz $\theta = [\theta_{ij}]$, de dimensão $n \times n$, em complemento a matriz de custos C . A entrada θ_{ij} é o nó predecessor do nó x_j no trajeto mínimo entre os nós x_i e x_j . A matriz θ é inicializada tal que $\theta_{ij} = x_{ij}$ para todo x_i e x_j .

Para o cálculo da equação utilizada pelo passo 4 do algoritmo de Floyd é introduzida a atualização da matriz θ como segue:

$$\theta_{ij} = \begin{cases} \theta_{kj}, & \text{se } (c_{ik} + c_{kj}) < c_{ij} \\ \theta_{ij}, & \text{se } (c_{ik} + c_{kj}) \geq c_{ij} \end{cases} \quad (2.9)$$

Ao final do algoritmo o trajeto mínimo pode ser obtido diretamente da matriz final θ . Assim, para obter o caminho mínimo entre quaisquer dois nós x_i e x_j , este caminho é dado pela seqüência de nós: $x_i, x_z, \dots, x_c, x_b, x_a, x_j$ onde $x_a = \theta_{ij}$, $x_b = \theta_{ia}$, $x_c = \theta_{ib}$, ..., $x_i = \theta_{iz}$

2.11 ANÁLISE DE AGRUPAMENTOS (*CLUSTERS*)

De acordo com Bráulio (2005), a Análise de agrupamentos consiste numa técnica exploratória cuja aplicação tem por objetivo a formação de grupos homogêneos de objetos (ou variáveis).

Os grupos são formados com base no cálculo das distâncias entre os itens, representados por vetores compostos por suas características, construindo-se uma matriz de distâncias e juntando os itens em grupos conforme suas proximidades. A reunião de itens similares em determinados grupos é importante, pois é freqüente a ocorrência de situações onde se deseja separar por uma determinada característica.

Esta técnica torna-se fácil de ser aplicada quando se trabalha com muitas variáveis, o que não ocorre neste trabalho, pois ocorre justamente o contrário, há somente uma variável (distância) para definir os grupos.

2.11.1 Medidas de Similaridade e Dissimilaridade

Em suas notas de aula Marques (2005) detalha que quando itens (unidades ou casos) são agrupados, a proximidade é usualmente indicada por uma espécie de distância.

As variáveis também são usualmente agrupadas com base nos coeficientes de correlação ou outras medidas de associação.

- Similaridade: quanto maior o valor observado, mais parecidos são os objetos.
- Dissimilaridade: quanto maior o valor observado, menos parecidos.

Quanto à distância, podem ser usadas algumas variações para comparação dos itens, onde a mais conhecida é a distância euclidiana. Seguem abaixo algumas delas:

Distância Euclidiana: Essa é, provavelmente, a mais conhecida e usada medida de distância. Ela é simplesmente a distância geométrica no espaço multidimensional. A distância entre dois vértices x e y é definida por:

$$d(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \quad (2.10)$$

Quadrado da Distância Euclidiana:

$$d(x, y) = \sum_{i=1}^p (x_i - y_i)^2 \quad (2.11)$$

Distância City Block (Manhattan):

$$d(x, y) = \sum_{i=1}^p |x_i - y_i| \quad (2.12)$$

Distância de Mahalanobis (distância estatística):

$$d(\underline{x}, \underline{y}) = \sqrt{\frac{(x_1 - y_1)^2}{s_1^2} + \dots + \frac{(x_p - y_p)^2}{s_p^2}} \quad (2.13)$$

Métrica de Minkowski:

$$d(x, y) = \sqrt[n]{\sum_{i=1}^p |x_i - y_i|^n} \quad (2.14)$$

2.11.2 Método de Agrupamento

Os métodos de agrupamentos podem ser hierárquicos ou não-hierárquicos, Braulio (2005) descreve este método como um método, onde, iniciam-se com tantos grupos quantos clientes (itens). Diversos itens semelhantes são agrupados primeiro, estes grupos iniciais são fundidos de acordo com as suas similaridades, eventualmente, relaxando no critério de similaridade os subgrupos vão se unindo à outros subgrupos até formar um grupo único.

O procedimento é o seguinte:

1. No início tem-se n grupos, sendo que cada grupo é formado por um único item; calcula-se a matriz simétrica de distâncias $n \times n$, $D = [d_{ij}]$ onde d_{ij} é a distância ou similaridade entre o item i e o item j .

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix} \quad (2.15)$$

Onde

$$d_{11} = d_{22} = \dots d_{nn} = 0 \quad (2.16)$$

2. Na matriz D , acha-se o par de grupos mais próximos (menor distância) e juntam-se estes grupos.
3. O novo grupo formado é denominado, por exemplo: (A, B) , se os grupos primitivos do par são A e B . Nova matriz de distâncias é construída, simplesmente apagando-se as linhas e colunas correspondentes aos grupos A e B e adicionando-se a linha e a coluna dadas pelas distâncias entre (AB) e os grupos remanescentes.
4. Repetem-se os passos 2 e 3 $n - 1$ vezes observando-se as identidades dos grupos que são agrupados.

Já o agrupamento não-hierárquico é uma técnica usada quando se deseja k grupos de itens. O método aglomerativo não-hierárquico mais usado é o algoritmo das k -médias. Este método é composto por 3 etapas:

1. Partição arbitrária dos itens em k grupos iniciais;
2. Re-alocar cada item n ao grupo cuja média (centróide) esteja mais próximo. Em geral é usada a Distância Euclidiana. O centróide é recalculado para o grupo que recebe novo item e para o grupo que perdeu algum item;
3. Repete-se a 2ª etapa até que não restem mais re-alocações a serem feitas.

Cada um dos métodos descritos realiza os agrupamentos por meio de ligações, sendo as mais comuns: Ligações simples (vizinho mais próximo), Ligações completas (vizinho mais distante), Método das Médias das Distâncias, Método do Centróide, Método de *Ward*.

2.12 AVALIAÇÃO DA FORMAÇÃO DOS AGRUPAMENTOS

Uma forma de avaliar a validade da informação gerada pela função ligação é compará-la com os dados originais da distância. Se o agrupamento é válido, a ligação dos objetos no agrupamento tem uma forte correlação com as distâncias entre objetos no vetor de distâncias. A função cofenética implementada no Matlab[®]7 compara esses dois conjuntos de valores e calcula sua correlação. A melhor solução para um agrupamento tem correlação igual a 1.

2.13 O ALGORITMO DAS P-MEDIANAS DE TEITZ E BART

O problema das p-medianas é um dos problemas estudados na teoria dos grafos que tem como objetivo localizar facilidades ao longo de uma rede viária, minimizando a soma das distâncias de cada um dos vértices à facilidade mais próxima, ponderada por um fator de demanda.

Esta heurística é baseada na substituição de vértices exaustivamente e seu objetivo é, a partir de uma solução inicial, melhorar o valor da função objetivo a cada iteração.

Para melhor compreensão do algoritmo faz-se necessária a definição de alguns conceitos básicos de acordo com Colombo (2001).

Denomina-se número de transmissão à soma das menores distâncias existentes entre o vértice v_i e todos os outros vértices de um grafo.

Colombo (2001) comenta ainda que de acordo com Christofides (1975) o número *out*-transmissão e *in*-transmissão podem ser definidos da seguinte forma:

1. Número de *out*-transmissão é a distância total a ser percorrida desde o vértice v a todos os outros vértices do grafo e, é dado por:

$$\sigma_t(v_i) = \sum_{v_j \in V} w_j \cdot d(v_i, v_j) \quad (2.17)$$

Onde:

$$d(v_i, v_j) = \text{menor distância entre } v_i \text{ e } v_j \quad (2.18)$$

$$w_j = \text{peso associado ao vértice } j \quad (2.19)$$

2. Número de *in*-transmissão é a distância total a ser percorrida a partir de todos os vértices do grafo até o vértice v_i e, é dado por:

$$\sigma_t(v_j) = \sum_{v_i \in V} w_i \cdot d(v_j, v_i) \quad (2.20)$$

Onde:

$$d(v_i, v_j) = \text{menor distância entre } v_j \text{ e } v_i \quad (2.21)$$

$$w_j = \text{peso associado ao vértice } j \quad (2.22)$$

São chamados de *out*-mediana e *in*-mediana de um grafo, respectivamente, aos vértices \bar{v}_0 e \bar{v}_t que satisfazem às seguintes condições:

$$\sigma_0(\bar{v}_0) = \min_{v \in V} [\sigma_0(v)] \quad (2.23)$$

$$\sigma_t(\bar{v}_t) = \min_{v \in V} [\sigma_t(v)] \quad (2.24)$$

Ou seja, \bar{v}_0 é *out*-mediana se, entre todos os vértices do grafo, produzir a menor distância total desde si próprio até todos os outros vértices: e \bar{v}_t é *in*-mediana se, entre todos os vértices do grafo, produzir a menor distância total desde todos os outros vértices até si próprio.

Generalizando estes conceitos, pode-se considerar V_p um subconjunto do conjunto de vértices V do grafo $G(V, A)$ que possui p elementos, isto é, $|V_p| = p$ e $V_p \subset V$.

Então para o problema de se encontrar p -medianas, define-se:

$$d(V_p, v_j) = \min_{v_j \in V_p} [d(v_i, v_j)] \quad (2.25)$$

e

$$d(v_j, V_p) = \min_{v_i \in V_p} [d(v_j, v_i)] \quad (2.26)$$

$$\sigma_0(V_p) = \sum_{v_j \in V} w_j \cdot d(V_p, v_j) \quad (2.27)$$

e

$$\sigma_t(V_p) = \sum_{v_j \in V} w_j \cdot d(v_j, V_p) \quad (2.28)$$

Finalmente, denomina-se um conjunto de p vértices de p -out-mediana ou p -in-mediana aos conjuntos \bar{V}_{p0} e \bar{V}_{pt} , respectivamente para os quais:

$$\sigma_0(\bar{V}_{p0}) = \min_{V_p \subset V} [\sigma_0(v_p)] \quad (2.29)$$

e

$$\sigma_0(\bar{V}_{pt}) = \min_{V_p \subset V} [\sigma_0(v_p)] \quad (2.30)$$

Ou seja, um conjunto é p -out-mediana se, entre todos os outros conjuntos de p vértices do grafo, produzir a menor distância total desde si próprio até todos os outros vértices, e é p -in-mediana se, entre todos os outros conjuntos de p vértices do grafo, produzir a menor distância total desde todos os outros vértices até si próprio.

Ainda de acordo com Colombo (2001) escolhem-se inicialmente, p vértices que formarão o conjunto S inicial, considerando uma aproximação do conjunto \bar{V}_p das p medianas (solução ótima para o problema). Sendo V o conjunto de todos os vértices do grafo $G(V, A)$, testa-se, se qualquer vértice $v_i \in (V - S)$ pode substituir um vértice $v_j \in S$ e produzir um novo conjunto S' de número de transmissão.

O algoritmo continua até obter-se um conjunto \bar{S} , onde nenhuma substituição de um vértice de \bar{S} por um vértice de $V - \bar{S}$ produz um número de transmissão menor.

ALGORITMO:

Passo 1: Selecione um conjunto S , com $|S| = p$, ou seja, formar uma solução inicial para p-medianas;

Passo 2: Rotule todos os vértices $v_i \notin S$ como não-analisados;

Passo 3: Enquanto existirem vértices não-analisados em $V - S$ faça:

3.1 - Selecione um vértice não analisado $v_i \in (V - S)$ e calcule a redução do número de transmissão para todo $v_j \in S$:

$$\Delta_{ij} = \sigma(S) - \sigma(S \cup \{v_i\} - \{v_j\}) \quad (2.31)$$

3.2 - Faça $\Delta_{ij0} = \max_{v_i \in S} \lfloor \Delta_{ij} \rfloor$;

3.3 - Se $\Delta_{ij0} > 0$ faça $S \leftarrow S \cup \{v_i\} - \{v_{j0}\}$ e rotule v_{j0} como analisado;

3.4 - Se $\Delta_{ij0} \leq 0$, rotule v_i como analisado;

Passo 4: Se durante a execução do passo 3 houver alguma modificação no conjunto S , volte ao passo 2. Caso contrário PARE e apresente o conjunto S como uma aproximação para a solução do problema das p-medianas.

2.14 O GPS

O GPS, Sistema de Posicionamento Global, é um sistema de geoposicionamento por satélites artificiais, baseado em transmissão e recepção de sinais de rádio-frequência em uma faixa muito alta (1.2 a 1.6 Giga-Hertz), em que uma constelação de 24 satélites distribuídos em seis planos orbitais localizados a 20.000 km de altitude, transmitem os sinais da Banda L e os rastreadores recebem e decodificam

estes sinais que são utilizados para determinar a distância entre cada satélite rastreado e a antena do geoposicionador (rastreador).

O GPS nasceu no início da década de 70, patrocinado pelo DoD (“*Department of Defense*” dos EUA) e desenvolvido pelo MIT “*Massachusetts Institute of Technology*” para aplicações militares.

Em 1993, a constelação GPS ficou completa, permitindo a exploração máxima do complexo de satélites-rastreadores para fins civis.

O advento do GPS revolucionou as técnicas de posicionamento levando o cidadão comum a utilizá-las das formas mais variadas. Muitas áreas já possuem-no inserido em suas atividades. Dos grandes trabalhos científicos aos momentos de lazer esta ferramenta está cada vez mais presente.

As aplicações crescem a cada dia: navegação aérea, marítima e terrestre, mapeamentos urbanos e rurais, agricultura de precisão, segurança contra furto de veículos, localização de incêndios, apoio na definição de trilhas, *rallys*, etc.

Os avanços tecnológicos permitem o surgimento de receptores cada vez menores e mais precisos. Por isso, não é exagero dizer que chegará o dia em que os relógios de pulso, além da hora, apresentarão coordenadas por meio do GPS.

3 METODOLOGIA UTILIZADA PARA RESOLUÇÃO DO PROBLEMA REAL

Este capítulo apresenta uma metodologia para solucionar o problema real descrito no capítulo I, de maneira diferente das metodologias usualmente praticadas na literatura.

Trata ainda de todas as etapas envolvidas na resolução do problema.

3.1 OBTENÇÃO DA LOCALIZAÇÃO GEOGRÁFICA DOS CLIENTES

Por uma questão administrativa da empresa e a inviável necessidade de dispor de recursos financeiros relativamente altos com combustível, tempo e mão-de-obra para a realização de um trabalho de campo de obtenção dos dados, neste trabalho optou-se por utilizar somente informações dos clientes que são visitados por vendedores da sede da Fertimourão.

Foram utilizadas informações dos agricultores constantes no banco de dados da Fertimourão como clientes da mesma, porém, com algumas exclusões, baseadas no percentual de compras efetuadas junto à empresa durante o ano de 2006.

Alguns clientes, independente de serem visitados, não costumam adquirir produtos nem serviços da empresa por estarem ligados a outras empresas do mesmo ramo.

As exclusões foram orientadas pela gerência da própria empresa, sendo imposto ainda um limite de 40 clientes para cada vendedor.

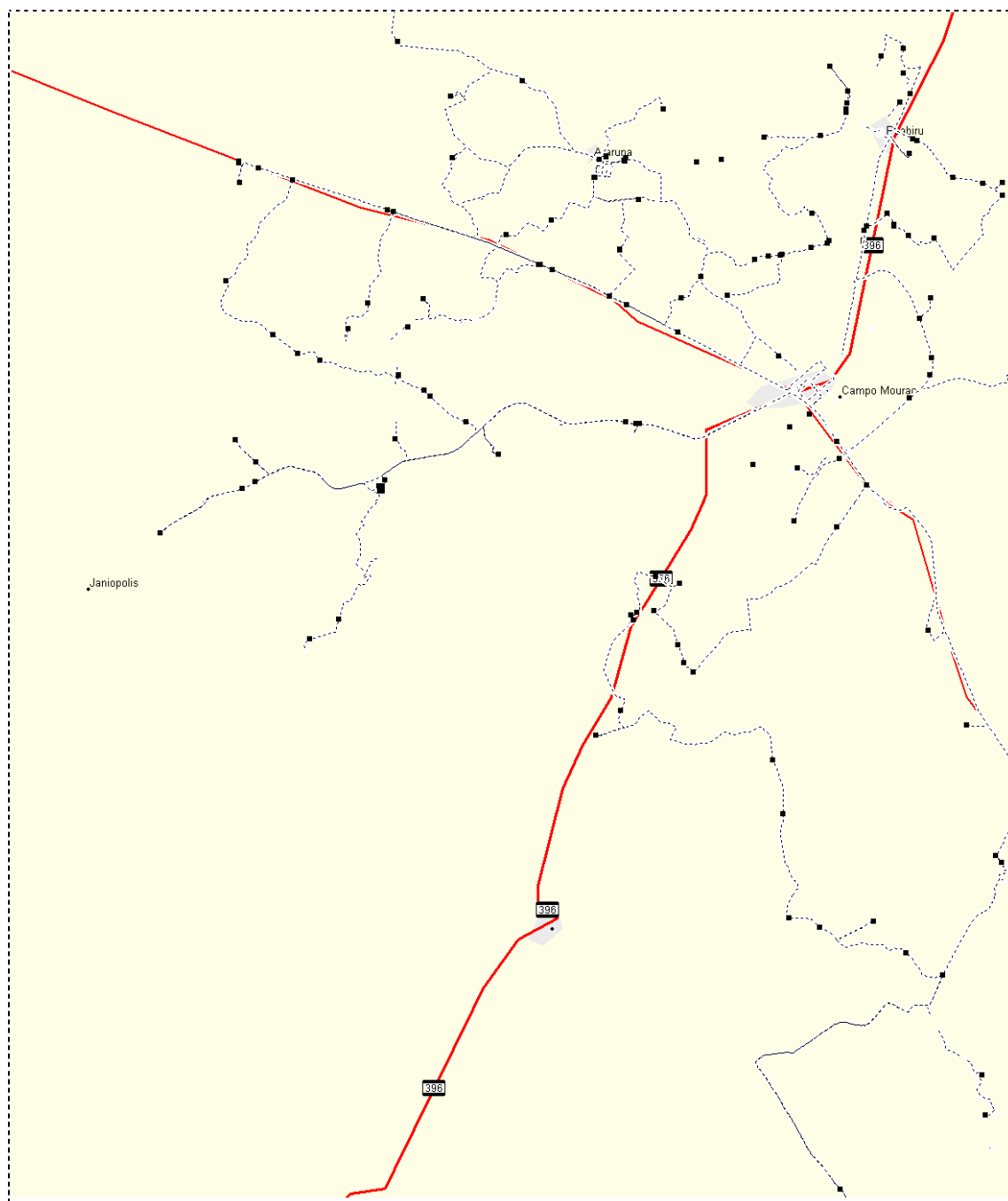
Após definido o número de clientes que formou o conjunto de pontos base deste trabalho, o próximo passo fundamentou-se na coleta das localizações geográficas e da malha viária, que foi realizada entre dezembro de 2006 e maio de 2007, com o auxílio de um aluno do curso de graduação em Matemática da Fecilcam (Faculdade Estadual de Ciências e Letras de Campo Mourão).

O aluno acompanhou as visitas rotineiras dos vendedores, levando consigo um GPS da marca Garmin (modelo Etrex-Vista) que atualmente é um aparelho bastante usado para este fim entre o meio empresarial por se tratar de um equipamento barato e com opções razoáveis de aquisição e manipulação de informações.

O equipamento utilizado possui uma taxa de erro de precisão na ordem de 5 a 20 metros. Esta taxa de erro foi desprezada neste trabalho por ser considerada insignificante para o resultado final.

Na figura 3.1, pode-se observar a região de abrangência da localização geográfica dos clientes em relação ao mapa do Estado do Paraná. Próximo à bifurcação entre as linhas vermelhas está localizada a cidade de Campo Mourão, na região centro oeste do estado do Paraná, onde é a sede da Fertimourão, local em que os vendedores iniciam e terminam seus trajetos todos os dias, os pontos pretos são as sedes ou residências dos clientes.

Figura 3. 1: Região de abrangência e Localização dos Clientes da Fertimourão



Esta região concentra as cidades de Campo Mourão, Araruna, Mamborê, Iretama, Roncador, Engenheiro Beltrão, Peabiru, Janiópolis, Farol, São Lourenço e proximidades de Cianorte. Alguns dos clientes não foram captados pelo zoom do software, principalmente na parte inferior do mapa.

Abaixo está a figura 3.2 com a região onde se localizam as cidades referidas acima:

Figura 3. 2: Localização das cidades com sedes de clientes



3.2 ELABORAÇÃO DA MATRIZ DE DISTÂNCIA ENTRE OS CLIENTES

Para aplicar-se qualquer algoritmo exato ou heurístico para resolução do PCV, do MPCV ou um problema de roteamento, deve-se conhecer as distâncias mínimas entre todos os pares de vértices ou pontos⁴. Estas distâncias, euclidianas ou reais podem ser obtidas através do algoritmo de Floyd (1962).

No caso particular da Fertimourão, não foi possível elaborar um banco de dados contendo todas as estradas possíveis entre todos os clientes da empresa, contudo

⁴ Neste caso os vértices ou pontos são os clientes da empresa ou uma bifurcação nas estradas

pôde-se obter pelo menos um caminho entre cada par de clientes permitindo assim, calcular as distâncias mínimas entre todos eles e gerar matriz de distâncias que serviu de base para o desenvolvimento deste trabalho.

3.2.1 Utilização do Algoritmo de Floyd

Encontrar a matriz de distâncias reais de forma manual somente com o auxílio da malha viária digitalizada, é praticamente impossível. Uma saída bastante rápida e interessante foi implementar do algoritmo de Floyd.

Este algoritmo reduz bastante o trabalho manual, que se concentra em acrescentar pontos em todos os locais onde as estradas possuem subdivisões e fornecer uma matriz contendo as distâncias entre todos os clientes e os pontos extras consecutivos. Tal matriz possui ainda o valor infinito para a distância entre dois pontos que não são consecutivos.

Após a alimentação do algoritmo com esta matriz, é calculada iterativamente a menor distância entre todos os pares de pontos e substituídos os valores infinitos pelo resultado dos cálculos.

Na malha viária que representa os caminhos entre os clientes da Fertimourão foram observados 56 pontos de subdivisão entre as estradas, além dos 154 pontos reais, totalizando uma matriz simétrica de dimensão 210×210 , na qual foi aplicado o algoritmo de Floyd implementado em Matlab[®]7 de acordo com o pseudocódigo constante no Anexo 1.

O referido pseudocódigo faz a leitura do arquivo contendo a matriz de distâncias entre todos os segmentos da malha viária e retorna uma matriz com as distâncias mínimas entre todos os clientes da Fertimourão.

3.3 ELABORAÇÃO DAS ROTAS UTILIZANDO HEURÍSTICAS

3.3.1 A Heurística de Clarke and Wright

De posse da matriz de distâncias, foi aplicada a Heurística de Clarke and Wright, implementada em Matlab[®]7 (Anexo 2), de acordo com os passos do algoritmo descrito no capítulo II, onde na primeira iteração foram geradas 153 rotas para atender individualmente cada cliente.

A cada iteração, as rotas foram se agrupando, observando as restrições do problema, de modo que ao final formou-se um conjunto de 32 rotas atendendo assim os 153 clientes da empresa.

A formação destas 32 rotas se deu devido às restrições impostas à heurística, referente ao tempo de atendimento de cada cliente e o tempo de locomoção do vendedor que somados não podem ultrapassar 6 horas diárias.

Sendo assim, cada rota não possui mais do que 6 ou 7 clientes, uma vez que o tempo médio de atendimento foi fixado pela empresa em 45 minutos.

Uma curiosidade é a uniformidade do comprimento médio de cada rota, que se deve justamente às restrições temporais do problema real.

3.3.1.1 Elaboração da Restrição de União das Rotas para a Heurística

Para a elaboração da restrição de união das rotas durante a execução da heurística, foi necessário o cálculo do tempo de locomoção à medida que a heurística gerava as rotas. Para isso utilizou-se a velocidade média de 40 km/h por se tratar de regiões de zona rural, e a fórmula da Física que relaciona a velocidade média, o tempo de locomoção e o tamanho da rota:

$$\text{Tempo de locomoção} = \frac{\text{Tamanho da rota}}{\text{Velocidade média}} \quad (3.1)$$

Depois se associou o tempo de locomoção, tempo de atendimento e o tamanho da rota para obter o tempo de trabalho gerado pelo cumprimento de cada rota diária.

$$\text{Tempo de Locomoção} + \text{Tempo de Atendimento} \leq \text{Jornada Diária} \quad (3.2)$$

$$\left(\frac{\text{Tamanho da Rota}}{\text{Velocidade Média}} \right) + (\text{Tamanho da Rota})(\text{Tempo de Atendimento}) \leq 6 \quad (3.3)$$

Desta forma, a expressão acima foi utilizada em um dos passos da heurística.

3.3.2 Heurística 2 – Opt

Como discutido anteriormente, a heurística de Clarke and Wright, é uma heurística de construção de rotas e que não garante a otimalidade da solução encontrada. Sendo assim foi implementada a heurística 2 – Opt (Anexo 3), que é uma heurística de melhoramento de solução, baseada na troca de vértices e que utilizou as rotas da solução de Clarke and Wright como ponto de partida.

O valor de “ k ” indica o número de vértices escolhidos para efetuar as trocas e neste trabalho foram implementadas trocas entre dois e três vértices somente, ou seja, 2 – Opt e 3 – Opt.

Implementações para trocas de quatro ou mais vértices, geralmente produzem melhores resultados, contudo segundo Nunes (1998) o tempo computacional é excessivamente alto, o que inviabiliza a sua implementação.

3.4 PROPOSTAS PARA UTILIZAÇÃO DO MODELO EXATO

Como discutido no capítulo II, a resolução de um problema de roteamento ou mesmo de um problema de múltiplos caixeiros viajantes através do modelo exato é

inviável, devido ao tempo computacional necessário para se obter a resposta ótima, principalmente se o número de vértices for relativamente alto.

Normalmente problemas com mais de 30 vértices tendem a não possuir solução em tempo computacional aceitável.

Ainda em relação ao modelo exato, a abordagem para resolução de um problema de roteamento proposta por Bodin (1983) parte de um problema de designação onde a quantidade de clientes é igual à quantidade de fábricas. Deste ponto de vista, a maior parte das variáveis que decidem as designações em determinada linha ou coluna de uma matriz receberão valor 0, pois somente a quantidade de variáveis igual à demanda da linha ou coluna é que serão designadas⁵.

Portanto, existem algumas variáveis ditas importantes para que o modelo possa realizar trocas e avaliar a opção que possui melhor valor para a função objetivo. As outras variáveis podem ser desprezadas, o que reduz consideravelmente o tempo computacional, pois limita o espaço de procura por soluções para o problema.

Com base em um critério sólido que garanta não infactibilizar o problema ou eliminar instâncias onde esteja contida a resposta ótima, ou ainda uma solução aceitável, estas variáveis ditas não importantes podem ser desprezadas reduzindo assim o tempo computacional e viabilizando a utilização desta variação do modelo exato.

Sendo assim, este trabalho propôs duas abordagens que associam um procedimento heurístico para subdividir um grande problema em grupos de problemas menores e ainda estabelecer em cada grupo menor, critérios que diminuam o número de variáveis e posteriormente roteirizar utilizando o modelo exato.

Também foi utilizada a técnica estatística da Análise de *Clusters* para transformar o problema maior em problemas menores, em substituição ao procedimento heurístico citado acima.

⁵ Ser designado significa que a variável de decisão recebe valor 1

Para a resolução do modelo exato foi utilizado o software LINGO, que segundo Souza (2005) é uma ferramenta computacional para modelagem e resolução de problemas lineares e não-lineares de otimização.

O processo de otimização consiste em tentar encontrar a melhor solução possível para um dado problema, usando técnicas de programação matemática, de forma a atingir o maior benefício/lucro ou gerar o menor custo/desperdício.

3.4.1 Teitz e Bart, Primeiro Critério de Eliminação de Variáveis e o Modelo Exato

A heurística de Teitz e Bart foi utilizada para apontar cinco clientes como referência, ou seja, encontrar cinco pontos estrategicamente distribuídos de forma que a soma das distâncias de todos os outros a um deles seja mínima.

Posteriormente resolveu-se um modelo de designação no LINGO, apresentado no Anexo 5, com o objetivo de agrupar os outros pontos aos 5 selecionados pelo algoritmo de Teitz e Bart.

Por fim, aplicou-se um critério para eliminar o máximo de variáveis dos subgrupos, baseado na premissa de que as maiores distâncias presentes na matriz dificilmente terão variáveis associadas com designação.

Foram feitos alguns testes para determinar o nível mais confiável de eliminação de variáveis, que convergissem para respostas razoáveis. Estes testes consistiam em utilizar um nível alto até tornar o problema infactível e posteriormente tomá-lo como referência para escolher um nível inferior.

Desta forma obtiveram-se alguns resultados que são apresentados e discutidos no capítulo V.

Para cada nível de eliminação foram encontrados vários mínimos locais para a função objetivo à tempos de processamento diferenciados e estes fatos também serão discutidos no próximo capítulo.

Vale lembrar que as respostas obtidas por este critério, não podem ser consideradas respostas ótimas.

3.4.2 Teitz e Bart, Segundo Critério de Eliminação de Variáveis (Modelo PL \rightarrow PLI) e o Modelo Exato

Ao analisarem-se as respostas de um modelo de roteamento, percebe-se que não é sempre que dois pontos mais próximos devem necessariamente estar ligados e, portanto, o critério de eliminar as variáveis onde as distâncias são maiores pode desprezar instâncias importantes no espaço de procura por soluções para o problema.

Diante desta percepção, da interpretação do problema de roteamento como um problema de designação que atenda às restrições de tempo, capacidade do veículo e outras mais que sejam particulares de cada problema, para elaboração do segundo critério de eliminação de variáveis foram levados em consideração os seguintes aspectos:

- Mesmo as variáveis que estejam relacionadas a distâncias maiores possuem chances de figurar na resposta ótima, ou ainda em maior chance, na melhor solução binária encontrada à um tempo computacional aceitável.
- Estas variáveis, só serão consideradas importantes em uma linha ou coluna caso existam distâncias na mesma linha ou coluna, razoavelmente menores, relacionadas a outras variáveis também com chances de figurarem na resposta final.
- Este conjunto de suposições é parte das condições que o modelo de roteamento utiliza para definir o grau de importância de cada variável.

Sendo assim, a idéia foi justamente propor um relaxamento na condição de variáveis inteiras à formulação exata do problema de roteamento e chama-lo de “PL inicial”.

Todas as variáveis que tinham valor diferente de zero, na primeira resolução do PL, eram consideradas do primeiro grau de importância, então elas eram zeradas ao resolver-se um outro PL para ser indicado outro grupo de variáveis do segundo grau de importância.

Unindo-se os dois grupos era rodado novamente o PI original à procura de uma solução.

Em todos os subproblemas, houve a necessidade de obter outras variáveis além das variáveis importantes do PL inicial, resolvendo outros PLs.

Este processo é iterativo, durando até que se tenha um bom conjunto de variáveis extraídas de cada resolução dos PLs, tornando assim o problema de roteamento factível e a um custo computacional bastante inferior ao problema original.

3.4.3 Técnica da Estatística (Análise de Clusters) e o Modelo Exato

Esta foi a última das metodologias a ser implementada, devido a uma característica bastante particular da mesma, normalmente os grupos ficam bem distintos, contudo não se tem o controle pleno do número de elementos em cada grupo.

Se comparada com as outras metodologias aqui apresentadas, esta característica quase inviabiliza a utilização da Análise de *Clusters* como ferramenta de auxílio na resolução de problemas de roteamento.

Contudo, depois de alguma reflexão, pôde-se concluir que de certa forma não há necessidade de possuir controle pleno da quantidade de elementos de cada grupo.

Para explicar melhor esta afirmação, considera-se uma situação onde existam 12 clientes para serem visitados e estejam disponíveis 4 veículos. De acordo com as propostas discutidas anteriormente, os clientes seriam separados em 4 grupos e em cada um, utilizar-se-iam os critérios discutidos.

Por consequência das características na Análise de *Clusters*, suponha que ao utilizar esta metodologia, fossem apresentados três grupos distintos (um com 3 clientes, outro com 6 e outro com mais 3).

Se os dois grupos de 3 clientes fossem roteirizados juntos, seria equivalente a uma roteirização de dois grupos de 6 clientes.

Testes que são discutidos nos próximos capítulos mostram, qual das técnicas possui melhor desempenho final, mas já adiantando o assunto para promover uma reflexão, ao serem roteirizados os dois grupos de 3 como se fosse um único, pode não haver prejuízo, pois a tendência é cada grupo receber um veículo diferente e formarem duas rotas independentes. No fim, acabam sendo geradas dentro dos dois grupos, as quatro rotas pedidas.

Portanto, resolveu-se o problema real utilizando esta técnica da seguinte forma:

- Primeiro verificou-se qual dos métodos de ligação mostrou-se mais eficiente para o caso em questão.
- Posteriormente realizou-se a clusterização com vários níveis para o parâmetro de entrada limitador da quantidade de grupos.
- Por fim, fizeram-se alguns ajustes manuais que aglutinaram pequenos grupos ate ter-se somente 6, e estes foram roteirizados segundo o modelo **PL** → **PLI** e o modelo exato.

4 IMPLEMENTAÇÃO COMPUTACIONAL

Neste capítulo são detalhados os modelos e as heurísticas utilizados para resolução do problema real.

Para a realização dos cálculos computacionais foram utilizados dois microcomputadores, o primeiro foi um Intel® Celeron® M 1.6 GHz 533MHz, 512MB de RAM e o outro um AMD Sempron (tm) 1.61 GHz, 533MHz, 512MB de RAM sendo que o Celeron se mostrou muito inferior no processamento dos mesmos cálculos que o Sempron.

4.1 O MODELO DE CLUSTERIZAÇÃO PARA DISTRIBUIR OS 154 CLIENTES EM 5 GRUPOS DISTINTOS

O modelo que segue foi elaborado com o simples objetivo de agrupar os outros clientes aos 5 selecionados pelo algoritmo de Teitz e Bart:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^n \sum_{j=1}^n \text{dist}_{i,j} x_{i,j} \\ \text{s.a :} \quad & \sum_{i=1}^5 \sum_{j=1}^n x_{i,j} \geq \text{ClusterMinimo}_i \\ & \sum_{i=1}^5 \sum_{j=1}^n x_{i,j} < \text{ClusterMaximo}_i \\ & \sum_{j=1}^n x_{i,j} = 1 \quad i = 1 \dots 5 \\ & \sum_{i=1}^n \sum_{j=1}^n x_{i,j} = n \\ & x_{i,j} \in \{0,1\} \quad i, j = 1 \dots n \end{aligned} \tag{4.1}$$

A variável de decisão $x_{i,j}$ indica em cluster i o cliente j será designado.

A expressão $\sum_{i=1}^n \sum_{j=1}^n \text{dist}_{i,j} x_{i,j}$ trata-se da função objetivo que propõe a designação de acordo com a distância total entre os clientes.

Logo em seguida vem $\sum_{i=1}^5 \sum_{j=1}^n x_{i,j} \geq \text{ClusterMinimo}_i$ e $\sum_{i=1}^5 \sum_{j=1}^n x_{i,j} < \text{ClusterMaximo}_i$,

sendo estes, dois conjuntos de restrições que definem o tamanho dos Clusters de acordo com uma faixa de variação entre os valores limites definidos pelas variáveis ClusterMinimo_i e ClusterMaximo_i .

Já o conjunto de restrições $\sum_{j=1}^n x_{i,j} = 1$ impede que um cliente seja designado para mais de um *Cluster* ao mesmo tempo.

Por último $\sum_{i=1}^n \sum_{j=1}^n x_{i,j} = n$ garante que nenhum cliente ficará sem ser designado para algum dos grupos, inclusive os clientes que são bases para a formação dos *Clusters*, pois a distância deles a eles mesmos é 0 e conseqüentemente são os primeiros a serem designados para seu próprio *Cluster*.

4.2 O MODELO UTILIZADO PARA ELIMINAÇÃO DE VARIÁVEIS

Para eliminar as variáveis de decisão consideradas sem importância para a construção das rotas, utilizaram-se várias metodologias.

Uma delas baseou-se em resolver o problema que segue e aproveitar sua resposta não inteira para apontar as variáveis de interesse.

$$\begin{aligned}
& \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^n \text{dist}_{i,j} x_{i,j} \\
& \text{s. a :} \\
& \quad x_{i,j} = 0 \quad \forall (i, j = 1 \dots n) \mid \text{dist}_{i,j} = 0 \\
& \quad \sum_{j=1}^n x_{i,j} = \text{Demanda}_i \quad \forall i = 1 \dots n \\
& \quad \sum_{i=1}^n x_{i,j} = \text{Demanda}_j \quad \forall j = 1 \dots n \\
& \quad \sum_{i=1}^n f_{i,j} - \sum_{i=1}^n f_{j,i} = \text{Demanda}_j \quad (4.2) \\
& \quad \sum_{j=1}^n x_{1,j} = \text{Demanda}_1 \\
& \quad \sum_{j=1}^n x_{j,1} = \text{Demanda}_1 \\
& \quad f_{i,j} = \text{CapVeic} \times x_{i,j} \quad i, j = 1 \dots n \\
& \quad \sum_{i=1}^n \sum_{j=1}^n x_{i,j} = \sum_{j=1}^n \text{Demanda}_j \\
& \quad x_{i,j} \in \mathbb{R} \quad i, j = 1 \dots n
\end{aligned}$$

Vale lembrar que para as duas metodologias que utilizaram a eliminação de variáveis, o modelo acima só foi utilizado a partir da 2ª iteração, pois, para obter o conjunto de variáveis importantes da primeira iteração, utilizou-se o próprio modelo exato de roteamento com relaxamento da condição de variáveis binárias, transformando-o num PL.

Algumas das restrições acima são as mesmas encontradas no PI para que a decisão das variáveis importantes seja influenciada por estas restrições que delimitam o problema na sua forma binária.

Exceto o primeiro e o último conjunto de restrições, que garantem que as variáveis que tiveram suas distâncias zeradas, não terão valor designado e que a soma de todas as variáveis designadas de forma “linear”⁶ será igual à soma das demandas.

⁶ Designada de forma linear quer dizer que qualquer valor diferente de 0, seleciona a variável como designada.

As expressões $\sum_{i=1}^n x_{i,j} = \text{Demanda}_j$ e $\sum_{j=1}^n x_{i,j} = \text{Demanda}_i$ representam o conjunto de restrições que direcionam o PL a designar variáveis em cada linha ou coluna, o suficiente para atender à sua demanda⁷, mesmo que não sejam variáveis inteiras.

4.3 O MODELO EXATO DE ROTEIRIZAÇÃO

O modelo exato foi formulado como segue:

$$\begin{aligned}
 \text{Min} \quad & \sum_{i=1}^n \sum_{j=1}^n \text{dist}_{i,j} x_{i,j} \\
 \text{s.a :} \quad & \\
 & \sum_{j=2}^n x_{i,j} = 1 \\
 & \sum_{i=2}^n x_{i,j} = 1 \\
 & \sum_{i=1}^n f_{i,j} - \sum_{i=1}^n f_{j,i} = \text{Demanda}_j \quad j \neq 1 \quad (4.3) \\
 & \sum_{j=1}^n x_{1,j} = \text{Demanda}_1 \\
 & \sum_{j=1}^n x_{j,1} = \text{Demanda}_1 \\
 & f_{i,j} = \text{CapVeic} \times x_{i,j} \quad i, j = 1 \dots n \\
 & x_{i,j} \in \{0,1\} \quad i, j = 1 \dots n
 \end{aligned}$$

A expressão $\text{Min} \sum_{i=1}^n \sum_{j=1}^n \text{dist}_{i,j} x_{i,j}$ é a função objetivo que mede o custo total das rotas. Por se tratar de um problema de minimização, as variáveis não nulas são escolhidas de forma que para atender todos os clientes, os trajetos percorridos sejam os menores possíveis.

⁷ Em nosso caso a demanda é sempre 1, porque cada cliente é visitado somente uma vez por dia.

A expressão $\sum_{j=2}^n x_{i,j} = 1$ é o primeiro conjunto de restrições do problema e garante que a partir de cada cliente sai somente um vendedor.

O conjunto de restrições dado pela equação $\sum_{i=2}^n x_{i,j} = 1$ limita cada cliente a receber a visita de um único vendedor.

Caso a abordagem deste problema não fosse tão particular, de forma que em cada dia um cliente é visitado uma única vez, o valor 1 de ambos os conjuntos, poderia ser substituído pela demanda de cada cliente, ou seja, pela quantidade de vezes que o cliente deve ser visitado.

Já a expressão $\sum_{i=1}^n f_{i,j} - \sum_{i=1}^n f_{j,i} = \text{Demanda}_j$ é um conjunto de restrições para a variável $f_{i,j}$ que unida à $f_{i,j} = \text{CapVeic} \times x_{i,j}$, controlam a formação de sub-rotas, cuja situação não é garantida pelos dois primeiros grupos de restrições.

A necessidade dessas expressões foi discutida no capítulo II onde Golden (1977) e Christofides (1975) propuseram duas formulações para resolver de forma exata o problema do caixeiro viajante, que é um caso particular do problema de roteamento. Ambos fazem referência a este grupo de restrições proibitivas.

Como a sede não foi incluída no primeiro e no segundo conjunto de restrições, quanto ao número de vendedores que chegam e saem dela, as expressões

$$\sum_{j=1}^n x_{j,1} = \text{Demanda}_1 \text{ e } \sum_{j=1}^n x_{1,j} = \text{Demanda}_1 \text{ fazem este papel.}$$

E por último, a condição de que a variável $x_{i,j}$ deve ser binária, ou seja, 0 caso o caminho do cliente i ao cliente j não faça parte de nenhuma alguma rota, e 1 caso contrário.

5 ANÁLISES DOS RESULTADOS

Neste capítulo serão apresentados os resultados obtidos por meio de cada uma das metodologias propostas, bem como realizadas algumas comparações entre as mesmas para analisar e explicar seus desempenhos.

5.1 RESULTADOS OBTIDOS UTILIZANDO HEURÍSTICAS

A primeira metodologia implementada foi a heurística de Clarke and Wright, que produziu os resultados da tabela abaixo.

Tabela 5. 1: Resultados Gerais de Acordo com a Heurística de Clarke and Wight

	FO	Tempo	Vel. Média	Tempo Atendimento
Problema 1	2383881 m	2 min	45 km/h	45 min
Total	2383881			

De acordo com a tabela acima, esta heurística requer um tempo computacional bem pequeno (apenas 2 minutos), com um valor para a função objetivo de 2383881 metros.

A tabela 5.2 refere-se ao conjunto de rotas apresentado pela heurística.

Tabela 5. 2: Conjunto de Rotas de Acordo com a Heurística de Clarke and Wright.

	Rota								Custo	Tamanho
Rota 01	54	01	98	102	126	54			89872	4
Rota 02	54	02	20	149	27	32	54		89664	5
Rota 03	54	03	70	121	140	122	54		65474	5
Rota 04	54	06	106	34	134	101	12	54	51366	6
Rota 05	54	08	85	87	16	105	104	54	65338	6
Rota 06	54	13	74	97	146	31	54		68590	5
Rota 07	54	17	66	80	148	54			102144	4
Rota 08	54	21	22	78	144	63	54		64602	5
Rota 09	54	25	111	56	117	128	124	54	55262	6
Rota 10	54	26	24	92	96	147	54		79156	5
Rota 11	54	28	29	71	42	129	51	54	65892	6
Rota 12	54	37	11	58	112	119	131	54	67663	6

Rota 13	54	38	43	115	10	141	54		103976	5
Rota 14	54	39	04	53	136	54			43962	4
Rota 15	54	41	52	75	154	152	125	54	64887	6
Rota 16	54	44	45	07	62	139	54		71442	5
Rota 17	54	48	94	135	54				114866	3
Rota 18	54	49	142	54					91226	2
Rota 19	54	61	69	33	77	89	54		65970	5
Rota 20	54	64	05	91	19	54			60500	4
Rota 21	54	65	35	72	143	88	54		81157	5
Rota 22	54	67	09	68	95	81	54		45467	5
Rota 23	54	73	15	100	113	120	54		86238	5
Rota 24	54	76	18	47	103	90	54		66473	5
Rota 25	54	79	30	57	99	54			93626	4
Rota 26	54	83	55	137	153	145	54		77692	5
Rota 27	54	84	114	59	116	54			63068	4
Rota 28	54	86	118	123	50	150	54		65744	5
Rota 29	54	93	82	127	54				83494	3
Rota 30	54	107	36	23	40	133	138	54	61634	6
Rota 31	54	108	60	132	110	54			86600	4
Rota 32	54	130	14	46	109	151	54		90836	5
Total									2383881	153

Na tabela acima, o número 54 refere-se à sede, ou seja, a Fertimourão e os outros números aos 153 clientes da empresa.

Nesta tabela pode-se observar que para atender todos os clientes, são necessárias 32 rotas, sendo que cada um dos 5 vendedores da empresa fica encarregado de cumprir um grupo⁸ delas que na prática é sua carteira de clientes.

Nota-se ainda que, cada rota não ultrapassou 6 clientes, com um custo aproximado de 80000 metros, pois elas são diárias, com tempo de atendimento de 45 minutos para cada cliente, o que limita bastante o tamanho da rota.

As rotas constantes na tabela 5.2 podem ser visualizadas no anexo 10.

Utilizando os resultados da tabela 5.2 como solução inicial, implementou-se a heurística 2 – *Opt*. A nova função objetivo, os novos custos para cada rota e as novas rotas pode ser observado nas tabelas 5.3 e 5.4.

⁸ Escolher os grupos de rotas da tabela 5.2 para cada vendedor não influencia nos custos finais, apesar dos algoritmos exatos (que discutiremos posteriormente) terem sido implementados com base em 5 grupos distintos de clientes.

De acordo com as tabelas 5.3 e 5.4, percebe-se mudanças mais significativas somente na quantidade de clientes atendidos em cada rota o que ocasionou uma redução na função objetivo em 81505 metros.

Na prática este comportamento da heurística está relacionado ao fato de que em algumas situações é mais compensador deixar um cliente para uma rota mais próxima a ele do que tentar equilibrar a quantidade de clientes atendidos em todas as rotas.

Os resultados das heurísticas, expostos até aqui, serão utilizados como parâmetro de análise das próximas metodologias propostas.

As rotas constantes na tabela 5.4 podem ser visualizadas no anexo 11.

5.2 RESULTADOS OBTIDOS UTILIZANDO O MODELO EXATO

Conforme discutido nos capítulos anteriores, foram propostas três metodologias diferentes para resolução do problema real, utilizando o modelo exato. Os resultados são apresentados nos tópicos seguintes.

5.2.1 Resultados Obtidos Utilizando Teitz e Bart, Critério Superior e o Modelo Exato

Esta metodologia consistiu na utilização do algoritmo de Teitz e Bart para dividir o problema original em 5 subproblemas, posteriormente foi aplicado um critério para eliminar variáveis, possibilitando a resolução do problema de roteamento por meio do modelo exato, o que foi feito e cujos resultados obtidos estão presentes nas tabelas seguem.

Tabela 5. 5: Resultados Gerais de Acordo Com Teitz e Bart e o Critério Superior.

	FO	Tempo	Variáveis	Restrições	Parâmetro
Problema 1	320012 m	10 min	435	929	15
Problema 2	396415 m	10 min	528	1189	16
Problema 3	421914 m	10 min	594	1189	18
Problema 4	583469 m	10 min	578	681	17
Problema 5	597149 m	10 min	522	610	18
Total	2318959				

Tabela 5. 6: Conjunto de Rotas de Acordo Com Teitz e Bart e o Critério Superior.

	Rota								Custo	Tamanho
Rota 01	54	11	148	133	40	16	105	54	52308	6
Rota 02	54	12	138	06	106	34	134	54	51366	6
Rota 03	54	59	116	35	88	75	54		64319	5
Rota 04	54	101	85	87	08	84	54		75356	5
Rota 05	54	119	112	58	114	131	36	54	76663	6
Rota 06	54	31	51	67	54				29264	3
Rota 07	54	39	04	01	98	96	09	54	111629	6
Rota 08	54	43	25	26	120	73	125	54	68699	6
Rota 09	54	68	95	130	38	27	32	54	51633	6
Rota 10	54	136	53	111	117	56	37	54	62274	6
Rota 11	54	144	63	115	107	104	54		72916	5
Rota 12	54	44	65	81	54				37990	3
Rota 13	54	66	62	55	137	153	145	54	78700	6
Rota 14	54	76	90	154	52	18	86	54	61418	6
Rota 15	54	109	14	46	103	47	80	54	100938	6
Rota 16	54	139	45	07	17	83	54		67882	5
Rota 17	54	151	152	41	72	143	23	54	74986	6
Rota 18	54	02	03	140	93	54			63430	4
Rota 19	54	15	127	147	92	82	113	54	163962	6
Rota 20	54	29	42	129	60	110	123	54	89004	6
Rota 21	54	118	146	132	97	71	24	54	101033	6
Rota 22	54	121	70	122	100	20	149	54	101078	6
Rota 23	54	150	74	13	108	50	54		64962	5
Rota 24	54	19	91	05	64	54			60500	4
Rota 25	54	30	49	78	21	142	141	54	175922	6
Rota 26	54	61	22	33	69	89	77	54	65984	6
Rota 27	54	94	99	135	48	79	54		115866	5
Rota 28	54	124	54						11262	1

Rota 29	54	128	28	126	102	57	10	54	167615	6
Total									2318959	153

As rotas constantes na tabela 5.6 podem ser visualizadas no anexo 12.

5.2.2 Resultados Obtidos Utilizando Teitz e Bart, Modelo PL → PLI e o Modelo Exato

A segunda metodologia de resolução do problema por meio do modelo exato também considerou a estratégia de utilizar o algoritmo de Teitz e Bart, contudo mudou-se o critério de eliminação de variáveis, denominado neste trabalho de modelo PL → PLI, seguindo rigorosamente a metodologia proposta no capítulo III obteve-se os resultados expostos nas tabelas 5.7 e 5.8.

Tabela 5. 7: Resultados Gerais de Acordo Com Teitz e Bart e o Critério PL → PLI.

	FO	Tempo	Variáveis	Restrições	Parâmetro
Problema 1	322486 m	10 min	217	305	4 Iterações
Problema 2	360963 m	10 min	377	477	6 Iterações
Problema 3	419334 m	10 min	252	352	4 Iterações
Problema 4	487987 m	10 min	387	490	6 Iterações
Problema 5	448255 m	10 min	222	310	4 Iterações
Total	2039025				

Tabela 5. 8: Conjunto de Rotas de Acordo Com Teitz e Bart e o Critério PL → PLI

	Rota								Custo	Tamanho
Rota 01	54	11	138	06	134	12	54		47566	5
Rota 02	54	58	84	114	116	59	36	54	67268	6
Rota 03	54	101	34	85	106	105	54		63270	5
Rota 04	54	131	112	119	35	88	75	54	72428	6
Rota 05	54	148	133	40	08	87	16	54	71954	6
Rota 06	54	09	130	38	54				36428	3
Rota 07	54	25	117	53	136	04	120	54	61800	6
Rota 08	54	32	27	26	96	73	54		63560	5
Rota 09	54	39	111	01	98	56	63	54	71165	6
Rota 10	54	51	67	31	68	95	125	54	46010	6

Rota 11	54	115	43	104	37	107	144	54	82000	6
Rota 12	54	62	55	137	153	23	54		82835	5
Rota 13	54	81	65	44	83	86	54		48555	5
Rota 14	54	90	47	152	18	151	76	54	66473	6
Rota 15	54	109	14	46	80	54			80728	4
Rota 16	54	145	139	07	45	66	17	54	68890	6
Rota 17	54	154	143	72	52	41	103	54	71853	6
Rota 18	54	50	150	110	60	71	54		77688	5
Rota 19	54	93	122	92	147	97	54		77500	5
Rota 20	54	100	149	20	113	15	02	54	98144	6
Rota 21	54	121	70	140	127	82	03	54	85198	6
Rota 22	54	129	132	42	29	24	54		71466	5
Rota 23	54	146	13	74	108	123	118	54	77991	6
Rota 24	54	28	54						4400	1
Rota 25	54	61	78	22	64	05	19	54	81538	6
Rota 26	54	79	128	124	54				59338	3
Rota 27	54	91	89	77	33	69	21	54	70696	6
Rota 28	54	135	99	94	30	57	48	54	132206	6
Rota 29	54	141	10	142	49	102	126	54	100077	6
Total									2039025	153

As rotas constantes na tabela 5.8 podem ser visualizadas no anexo 13.

5.2.3 Resultados Obtidos Utilizando a Estatística, Modelo PL → PLI e o Modelo Exato

A última metodologia procurou utilizar o critério de eliminação de variáveis (PL → PLI) que apresentou melhor desempenho associando-o a um meio mais eficaz para divisão do grupo principal em grupos menores. Para isto implementou-se a técnica estatística da Análise de *Clusters*, conforme discutido anteriormente.

Esta técnica não levou em conta as mesmas premissas⁹ que as metodologias anteriores, sendo que a subdivisão resultou em 6 subgrupos e cujos resultados alcançados podem ser observados nas tabelas abaixo.

⁹ O termo “premissas” refere-se a divisão dos clientes em 5 grupos, os quais representavam a carteira de cada um dos 5 vendedores.

Tabela 5. 9: Resultados Gerais de Acordo Com a Estatística e o Critério PL → PLI.

	FO	Tempo	Variáveis	Restrições	Parâmetro
Problema 1	427303 m	10 min	169	260	4 iterações
Problema 2	211069 m	10 min	112	173	3 iterações
Problema 3	370541 m	10 min	179	276	4 iterações
Problema 4	332026 m	10 min	135	208	3 iterações
Problema 5	186995 m	10 min	150	232	3 iterações
Problema 6	140891 m	10 min	144	223	4 iterações
Total	1668825				

Tabela 5. 10: Conjunto de Rotas de Acordo Com a Estatística e o Critério PL → PLI.

Rota									Custo	Tamanho
Rota 01	54	01	98	126	102	135	48	54	98449	6
Rota 02	54	33	69	21	22	144	78	54	66920	6
Rota 03	54	66	07	153	137	55	62	54	78700	6
Rota 04	54	79	61	54					42242	2
Rota 05	54	89	77	05	64	19	91	54	73236	6
Rota 06	54	145	139	45	54				67756	3
Rota 07	54	14	65	41	52	109	54		81540	5
Rota 08	54	18	154	143	88	72	54		50837	5
Rota 09	54	103	46	90	75	47	54		47810	5
Rota 10	54	151	17	152	44	54			30882	4
Rota 11	54	28	32	27	124	54			24862	4
Rota 12	54	37	107	11	12	104	125	54	37338	6
Rota 13	54	39	111	117	56	128	25	54	44693	6
Rota 14	54	53	04	136	49	142	99	54	97230	6
Rota 15	54	63	57	30	94	10	141	54	150834	6
Rota 16	54	115	38	43	54				15584	3
Rota 17	54	20	149	100	113	15	82	54	115629	6
Rota 18	54	83	68	09	95	81	130	54	47058	6
Rota 19	54	86	80	76	31	51	54		59952	5
Rota 20	54	127	150	50	118	123	67	54	109387	6
Rota 21	54	26	13	42	73	24	120	54	74992	6
Rota 22	54	70	121	02	140	54			9272	4
Rota 23	54	74	108	110	60	132	129	54	45269	6
Rota 24	54	93	54						6738	1
Rota 25	54	96	146	03	97	29	71	54	26986	6
Rota 26	54	122	92	147	54				23738	3
Rota 27	54	16	36	08	06	87	85	54	27390	6
Rota 28	54	84	58	59	34	116	114	54	29958	6
Rota 29	54	105	138	40	54				40454	3

Rota 30	54	106	23	134	101	54			8016	4
Rota 31	54	133	112	119	131	35	148	54	35073	6
Total									1668825	153

As rotas constantes na tabela 5.10 podem ser visualizadas no anexo 14.

5.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS

Normalmente não se utiliza na bibliografia a formulação exata para resolução de problemas de roteamento, devido ao tempo computacional elevado para se atingir a resposta ótima, ou mesmo, uma resposta viável. Observa-se nas tabelas 5.5, 5.7 e 5.9, que todas as três metodologias aqui propostas se revelaram adaptadas em relação à esta questão, pois, obtiveram resultados razoáveis em cerca de uma hora.

Se comparadas com as heurísticas de Clarke and Wright, essas metodologias são inferiores no que diz respeito ao tempo de processamento e não se justificaria o uso das mesmas se as rotas apontadas não fossem melhores que no caso das heurísticas.

A metodologia do critério superior mostrou-se equivalente à heurística de Clarke and Wright, apontando um resultado 64922 m menor, porém quando aplicado a heurística de melhoramento 2 – *Opt* esta diferença reduziu-se a praticamente zero (os valores podem ser observados nas tabelas 5.3 e 5.5).

A metodologia do modelo PL → PLI mostrou-se ainda melhor que as duas anteriores, promovendo um decréscimo de 279934 m nas distâncias totais percorridas. Esta melhora justifica-se pelo fato das variáveis a serem desprezadas terem sido escolhidas não só por estarem relacionadas a distâncias menores, mas pelas chances maiores de figurarem nas rotas.

Diante de todos os resultados obtidos, o melhor conjunto de rotas foi apontado pela última metodologia, que utilizou o modelo PL → PLI, mas mudou os meios para determinar a divisão dos clientes em subgrupos menores, onde foi utilizada a técnica estatística *Análise de Clusters* em substituição do algoritmo de Teitz e Bart. Os

resultados apontaram uma distância total percorrida de 1668825 m, ou seja, cerca de 630000 m menor que as heurísticas e 370000 m melhor que os outros critérios.

Outro fator importante para se ressaltar, está relacionado ao número de variáveis e restrições das metodologias que utilizaram o modelo exato, percebe-se através das tabelas 5.5, 5.7 e 5.9 que os critérios que utilizaram Teitz e Bart, possuem soluções inferiores apesar de considerarem um número maior de variáveis e restrições, que a metodologia envolvendo a Análise de *Clusters*.

Conclui-se então, que não é a quantidade de variáveis a serem consideradas que melhora ou piora o resultado final e sim sua qualidade.

Um dos principais motivos para que tenham sido implementadas as heurísticas de Clarke and Wright e *2_Opt*, relaciona-se à possibilidade de comparação com as outras metodologias propostas. Conforme discutido no capítulo II, existem várias outras formas de resolução de problemas de roteamento que não foram implementadas aqui e para que haja uma maior validade dos resultados faz-se presente a tabela abaixo que expõem a resolução linear para o problema com todos os clientes da Fertimourão.

Tabela 5. 11: Resultados Gerais de Acordo Com a Formulação Exata Para Variáveis Não Inteiras.

	FO (PL)	Tempo	Variáveis	Restrições
Problema 1	1103938 m	38 seg.	47278	24025
Total	1103938			

A tabela acima fornece um parâmetro mínimo para a função objetivo, uma vez que o método Banch-and-Bound, utilizado pelo LINGO, parte deste problema de Programação Linear para buscar soluções inteiras piorando a mesma. Qualquer solução para o problema real terá como função objetivo um valor maior que o exposto acima.

Diante das condições apresentadas na tabela 5.11 e da discussão do parágrafo anterior, pode-se concluir que o valor das distâncias totais percorridas, apresentado na

tabela 5.9, está muito próximo da função objetivo da tabela 5.11, o que implica em afirmar que esta metodologia apresentou um excelente desempenho.

6 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Ao término do presente trabalho pôde-se afirmar, fundamentado nos resultados obtidos, que é possível utilizar uma variação do modelo exato de roteamento para a resolução de problemas reais com a possibilidade de livrar-se da questão do tempo computacional.

Contudo, esta abordagem do problema não garante a otimalidade da resposta obtida, que inclusive não foi o foco deste trabalho, o qual almejou comparar os resultados obtidos com estas adaptações aos resultados normalmente encontrados quando se utiliza as técnicas praticadas na literatura para resolução de problemas como este.

Conclui-se ainda, que se as rotas indicadas por este trabalho forem praticadas, proporcionará ao setor de vendas da empresa um melhor desempenho por reduzirem os gastos mensais, sistematizarem a forma de trabalho, garantir maior agilidade no atendimento e a certeza de uma visita a cada cliente ao menos uma vez por mês.

Como discutido anteriormente, a utilização do modelo exato em problemas maiores, geralmente é inviável, contudo, indica-se para trabalhos futuros a exploração dos critérios apresentados aqui por meio de melhores formas de se obter bons conjuntos de variáveis para aplicar o roteamento sem considerar a totalidade do problema, e ainda atingindo bons resultados.

Outro ponto a ser explorado relaciona-se aos testes com situações que utilizem distâncias reais, comparando os resultados da roteirização com práticas que considerem, por exemplo, a distância euclidiana, obtendo parâmetros que estimem as diferenças.

A sugestão anterior é bastante útil, uma vez que dispor de toda a malha viária em problemas envolvendo rotas de veículos, é um grande problema e caso existam os tais parâmetros para zonas rurais, poupa-se muito tempo e gastos com aquisição de informações.

REFERÊNCIAS

BELTRAMI E. L. e BODIN L. D. **Networks and vehicle routing for municipal waste collection.** p.65-94.1974

BRANCHINI, Rodrigo Moretti. **Busca Tabu Para o Problema de Roteamento Dinâmico de Veículos com Janelas de Tempo.** Dissertação de Mestrado. Universidade Estadual de Campinas – UNICAMP. Faculdade de Engenharia Elétrica e de Computação, 2005.

BRAULIO, S.N. **Proposta de uma Metodologia para Avaliação de Imóveis Urbanos Baseado em Métodos Estatísticos Multivariados.** Dissertação de Mestrado, Universidade Federal do Paraná – UFPR, 2005.

BODIN L., GOLDEN B., ASSAD A., BALL M. **Routing and Scheduling of vehicles and crews:the state of the art.** Special Issue. England:Pergamon Press, 1983.

CHONG, Y.N. **Heuristic Algorithms For Routing Problems.** Ph.D. thesis, School of Mathematics and Statistics, Curtin University of Technology, 2001.

CLARKE, G. WRIGHT, J.W. **Scheduling of Vehicles from a Central Depot to a Number of Delivery Points.** Operations Research, v. 12, p.568-581, 1964.

COLOMBO, J.A. **Localização e Roteamento para Serviços de Atendimento Emergencial – O caso da Segurança Eletrônica.** Dissertação de Mestrado, Universidade Federal do Paraná – UFPR, 2001.

COSTA, D.M.B. **Aplicações de Algumas Técnicas da Pesquisa Operacional na Otimização dos Serviços Postais.** Dissertação de Mestrado, Universidade Federal do Paraná – UFPR, 1997.

FLOYD, R.W. **Algorithm 97:** shortest path. Communications of the Association for Computing Machinery, v.5, p.345, 1962.

GANHOTO, M.A. **Busca Tabu Aplicada ao Problema de Roteamento Periódico de Veículos.** Dissertação de Mestrado, Universidade Estadual de Campinas – UNICAMP. Instituto de Computação, 2004.

GARVIN W. W., GRANDALL J. B. and SPELLMAN R. A.. **Applications of Linear Programming in the Oil Industry**. Management Science, p.407-430, 1957.

GOLDBARG, M.C. LUNA, H.P.L. **Otimização Combinatória e Programação Linear**: modelos e algoritmos. Rio de Janeiro, Campus, 2000.

GOLDEN, B.L. MAGNANTI, T.L. NGUYEN, H.Q. **Implementing Vehicle routing algorithms**. Networks, v.7, p.113-148, 1977.

<http://aquarius.ime.eb.br/~webde6/gps.html>. Acesso em 04/11/2006.

LIN, S. KERNIGHAN, B.W. **An Effective Heurist Algorithm for the Traveling Salesman Problem**. Op. Res., v.21, p.498-516, 1973.

LIN, Shen. **Computer Solutions of the Traveling Salesman Problem**. The Bell System Technical Journal, v.44, 1965.

MARQUES, J.M. **Notas de Aula Professor Jair Mendes Marques**. Universidade Federal do Paraná – UFPR, 2005.

MATSUMOTO E.Y. **Matlab®7**: Fundamentos. 1ed. São Paulo, Érica, 2004.

MORTATI, C.F. **Busca Tabu Aplicada ao Problema de Roteamento Periódico de Veículos**. Dissertação de Mestrado. Universidade Estadual de Campinas – UNICAMP. Faculdade de Engenharia Elétrica e de Computação, 2005.

MULDER, S.A.; WUNSCH II, D.C. **Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks**. Neural Networks, v. 16, n. 5-6, p. 827-832, jun./jul. 2003.

NOVAES , A. G. **Using Geographical Information Systems in Conjunction with vehicle Routing Problems**. The International Federation of Operational Research Societies, 14th Triennial Conference, Canadá, 1996.

NOVAES, A. G. **Metric Relations And Approximations in Logistics Networks**. Proceedings of 7th Annual ANPET Meeting, Brazilian Association of Research and Teaching on Transportation, Brazil, v2, pg.580-591, 1993.

NUNES, L.F. **Algoritmos Genéticos Aplicados na Abordagem de um Problema Real de Roteirização de Veículos**. Dissertação de Mestrado, Universidade Federal do Paraná – UFPR, 1998.

RUSSEL R. A. e IGO W. **An assignment routing problem**, *Networks*. 1979.

SIQUEIRA, P.H. **Uma Nova Abordagem na Resolução do Problema do Caixeiro Viajante**. Tese de Doutorado, Universidade Federal do Paraná – UFPR, 2005.

SOUZA, L.V. **Técnicas de Roteirização de Veículos Aplicadas ao Transporte Escolar**. Dissertação de Mestrado, Universidade Federal do Paraná – UFPR, 1997.

SOUZA, M.J.F. et all. **Notas de Aula de Pesquisa Operacional Aplicada à Mineração**. Departamento de Computação Instituto de Ciências Exatas o Biológicas. Universidade Federal de Ouro Preto, 2005.

ANEXOS

LISTA DE ANEXOS

Anexo 1: Implementação Computacional em Matlab [®] 7 do Algoritmo de Floyd	70
Anexo 2: Implementação Computacional em Matlab [®] 7 da Heurística de Clarke and Wright	71
Anexo 3: Implementação Computacional em Matlab [®] 7 da Heurística 2- <i>Opt</i>	73
Anexo 4: Implementação Computacional em Matlab [®] 7 do Algoritmo de Teitz e Bart	75
Anexo 5: Implementação Computacional em LINGO do Modelo de Clusterização ...	76
Anexo 6: Implementação Computacional em LINGO do Critério Superior.....	77
Anexo 7: Implementação Computacional em LINGO do Modelo PL → PLI.....	78
Anexo 8: Implementação Computacional em Matlab [®] 7 da Análise de <i>Clusters</i>	79
Anexo 9: Implementação Computacional em LINGO do Modelo Exato de Roteamento	81
Anexo 10: Visualização gráfica das rotas indicadas na tabela 5.2	82
Anexo 11: Visualização gráfica das rotas indicadas na tabela 5.4	83
Anexo 12: Visualização gráfica das rotas indicadas na tabela 5.6	84
Anexo 13: Visualização gráfica das rotas indicadas na tabela 5.8	85
Anexo 14: Visualização gráfica das rotas indicadas na tabela 5.10.....	86

Anexo 1: Implementação Computacional em Matlab®7 do Algoritmo de Floyd

```
% Passo I
MatrizFloyd=xlsread('Floyd','B:HC');
n,n]=size(MatrizFloyd);

%Passos II, III e IV
for k=1:n
    Total=sum(sum(MatrizFloyd))
    for i=1:n
        for j=1:n
            if MatrizFloyd(i,k)~=1000000
                if MatrizFloyd(k,j)~=1000000
                    MatrizFloyd(i,j)=min(MatrizFloyd(i,j),MatrizFloyd(i,k)
                        +MatrizFloyd(k,j));
                end
            end
        end
    end
end

% Passo V
xlswrite('Floyd',MatrizFloyd,'Real2','B2');
```

Anexo 2: Implementação Computacional em Matlab® 7 da Heurística de Clarke and Wright

```
% Passo I (selecionar o vértice sede)
Sd=input('Inserir sede:');
T=input('Tempo de Atendimento(hrs):');
V=input('Velocidade Media(Km/h):');

% Passo II (criar m-1 rotas do tipo: sede - vértice i - sede)
[m,n]=size(Dados);
for i=1:m
    Dados(i,i)=0;
end
Rotas=[];
Economia=[];
for i=1:m
    if i~=Sd
        Rotas=[Rotas;Sd i];
    end
end

% Passo III (criar a lista das economias para a sede selecionada)
for i=1:m
    for j=i+1:m
        if i~=j&i~=Sd&j~=Sd
            Economia=[Economia;Dados(i,Sd)+Dados(Sd,j)-Dados(i,j),i ,j];
        end
    end
end

% Passo IV (ordenar a lista de economias de forma decrescente)
[l,k]=size(Economia);
for i=1:l
    for j=1:l
        if Economia(i,1)>Economia(j,1)
            C=Economia(i,:);
            Economia(i,:)=Economia(j,:);
            Economia(j,:)=C;
        end
    end
end

% Passo V (Unir rotas minimizando os custos e respeitando os limites )
RotasFinal=Rotas;
for i=1:l
    [a,b]=size(Rotas);
    [L,C]=find(Rotas(:,')==Economia(i,2));
    [H,G]=find(Rotas(:,')==Economia(i,3));
    [J,K]=find(Rotas(H,2:b)>0);
    tam=size(K);
    d=size(find(Rotas(L,:)>0));
    e=size(find(Rotas(H,:)>0));
    Rotas=RotasFinal;
    [l,c]=size(RotasFinal);
    Custos=zeros(1,1);
    for p=1:l
        for q=1:c-1
            if RotasFinal(p,q)>0 & RotasFinal(p,q+1)>0
                Custos(p,1)=[Custos(p,1)+Dados(RotasFinal(p,q),RotasFinal(p,q+1))];
            end
        end
    end
end
```

```

end
if RotasFinal(p,q+1)==0 & RotasFinal(p,q)~=0
    Custos(p,1)=[Custos(p,1)+Dados(RotasFinal(p,q),RotasFi
nal(p,1))];
end
if q==c-1 & RotasFinal(p,q+1)~=0
    Custos(p,1)=[Custos(p,1)+Dados(RotasFinal(p,q+1),Rotas
Final(p,1))];
end
end
end
for j=1:tam(2)
    if Rotas(H,G)>0 & L~=H & (Custos(L,1)+Custos(H,1))/(1000*V)+T*(d(2)+e(2)-
2)<=8
        Rotas(L,d(2)+j)=Rotas(H,2+j-1);
        if j==tam(2)
            Rotas(H,:)=[];
            RotasFinal=Rotas;
        end
    end
end
end

% Restaurando custos e apresentando o Resultado no prompt do Matlab
[l,c]=size(Rotas);
Custos=zeros(l,1);
for i=1:l
    for j=1:c-1
        if Rotas(i,j)>0 & Rotas(i,j+1)>0
            Custos(i,1)=[Custos(i,1)+Dados(Rotas(i,j),Rotas(i,j+1))];
        end
        if Rotas(i,j+1)==0 & Rotas(i,j)~=0
            Custos(i,1)=[Custos(i,1)+Dados(Rotas(i,j),Rotas(i,1))];
        end
        if j==c-1 & Rotas(i,j+1)~=0
            Custos(i,1)=[Custos(i,1)+Dados(Rotas(i,j+1),Rotas(i,1))];
        end
    end
end
end
clc
RotasFinal
CustoTotal=sum(Custos);
CustoTotal

```

Anexo 3: Implementação Computacional em Matlab®7 da Heurística 2 – Opt

```
% Passo I (Obtenção de um Circuito Hamiltoniano)
obter através do Clarke.m

% Passo II (Troca de 2 arestas)
teste=[];
Sd=input('Qual a Sede?');
[l,c]=size(Rotas);
RotasFinal=Rotas;
CustoTemp=zeros(1,1);
for i=1:l
    for j=1:c-1
        if Rotas(i,j)>0 & Rotas(i,j+1)>0
            CustoTemp(i,1)=[CustoTemp(i,1)+Dados(Rotas(i,j),Rotas(i,j+1))];
        end
        if Rotas(i,j+1)==0 & Rotas(i,j)~=0
            CustoTemp(i,1)=[CustoTemp(i,1)+Dados(Rotas(i,j),Rotas(i,1))];
        end
        if j==c-1 & Rotas(i,j+1)~=0
            CustoTemp(i,1)=[CustoTemp(i,1)+Dados(Rotas(i,j+1),Rotas(i,1))];
        end
    end
end
CustoFinal=CustoTemp;
CustoInicial=CustoTemp;
for i=1:l
    for j=1:c
        for k=1:l
            for m=1:c
                if Rotas(i,j)-Sd~=0 & Rotas(k,m)-Sd~=0 & Rotas(i,j)~=0 &
Rotas(k,m)~=0 & i~=k
                    teste=[teste;i,j,k,l,m,n];
                    Rotas0=Rotas;
                    Itemp=Rotas(i,j);
                    Rotas0(i,j)=Rotas(k,m);
                    Rotas0(k,m)=Itemp;
                    CustoTemp=zeros(1,1);
                    for r=1:l
                        for s=1:c-1
                            if Rotas0(r,s)>0 & Rotas0(r,s+1)>0
                                CustoTemp(r,1)=[CustoTemp(r,1)+Dados(Rot
as0(r,s),Rotas0(r,s+1))];
                            end
                            if Rotas0(r,s+1)==0 & Rotas0(r,s)~=0
                                CustoTemp(r,1)=[CustoTemp(r,1)+Dados(Rot
as0(r,s),Rotas0(r,1))];
                            end
                            if s==c-1 & Rotas0(r,s+1)~=0
                                CustoTemp(r,1)=[CustoTemp(r,1)+Dados(Rot
as0(r,s+1),Rotas0(r,1))];
                            end
                        end
                    end
                    if sum(CustoTemp)<sum(CustoFinal)
                        CustoFinal=CustoTemp;
                        RotasFinal=Rotas0;
                        Rotas=Rotas0;
                    end
                end
            end
        end
    end
end
```

```

if Rotas(i,j)~=0 & Rotas(k,m)~=0 & i==k
    teste=[teste;i,j,k,l,m,n];
    Rotas0=Rotas;
    Itemp=Rotas(i,j);
    Rotas0(i,j)=Rotas(k,m);
    Rotas0(k,m)=Itemp;
    CustoTemp=zeros(l,1);
    for r=1:l
        for s=1:c-1
            if Rotas0(r,s)>0 & Rotas0(r,s+1)>0
                CustoTemp(r,1)=[CustoTemp(r,1)+Dados(Rot
                    as0(r,s),Rotas0(r,s+1))];
            end
            if Rotas0(r,s+1)==0 & Rotas0(r,s)~=0
                CustoTemp(r,1)=[CustoTemp(r,1)+Dados(Rot
                    as0(r,s),Rotas0(r,1))];
            end
            if s==c-1 & Rotas0(r,s+1)~=0
                CustoTemp(r,1)=[CustoTemp(r,1)+Dados(Rot
                    as0(r,s+1),Rotas0(r,1))];
            end
        end
    end
    if sum(CustoTemp)<sum(CustoFinal)
        CustoFinal=CustoTemp;
        RotasFinal=Rotas0;
        Rotas=Rotas0;
    end
end
end
end
end

% Apresentar o Resultado
RotasFinal
Melhora=sum(CustoInicial-CustoFinal)
CustoTotal=sum(CustoFinal)

```

Anexo 4: Implementação Computacional em Matlab[®]7 do Algoritmo de Teitz e Bart

```
%Define a quantidade de pontos, de grupos e o grupo inicial
n=input('Quantas Cidades?');
g=input('Quantos grupos?');
Cluster=input('Entre com a matriz Cluster Inicial:');
for i=1:n
    Dados(i,i)=1000000;
end

% Executa o Passo I do Algoritmo de Teitz e Bart
NumTrans=sum(min(Dados(Cluster(1,:),:)))-
sum(min(Dados(Cluster(1,:),Cluster(1,:))));

% Executa o Passo II do Algoritmo de Teitz e Bart
ClusterFinal=Cluster;
NumTransTemp=10000000;
H=0;
while H<20
    H=H+1;
    for i=1:n
        for k=1:g
            if i~=Cluster(1,:)
                Cluster2=Cluster;
                Cluster2(1,k)=i;
                NumTransTemp=sum(min(Dados(Cluster2(1,:),:)))-
sum(min(Dados(Cluster2(1,:),Cluster2(1,:))));
            end
            if NumTransTemp<NumTrans
                NumTrans=NumTransTemp;
                ClusterFinal(1,k)=i;
                Cluster(1,k)=i;
            end
        end
    end
end

% Monta a Matriz para o LINGO Gerar os Clusters com base nas Sementes do
ClusterFinal
[w,m]=size(ClusterFinal);
Base=[];
for i=1:m
    Base=[Base,Dados(:,ClusterFinal(1,i))];
end
Base=[ClusterFinal;Base];
xlswrite('arquivo',Base,'Real','B1');
```

Anexo 5: Implementação Computacional em LINGO do Modelo de Clusterização

MODEL:

TITLE: Clusterizacao;

SETS:

Cidades / @OLE('E:\Metodologia 4 - Teitz, Guillet & ModeloExato\arquivo',
'Cidades') /:Visita;
Cluster /@OLE('E:\Metodologia 4 - Teitz, Guillet &
ModeloExato\arquivo','Clusters')/:CapCluster;
matriz(cidades,Cluster):dist,x;

ENDSETS

DATA:

!Importa os dados presentes no arquivo MatrizDados.xls;
dist= @OLE('E:\Metodologia 4 - Teitz, Guillet & ModeloExato\arquivo',
'dist');
CapCluster= @OLE('E:\Metodologia 4 - Teitz, Guillet & ModeloExato\arquivo',
'Capacidade');
Visita= @OLE('E:\Metodologia 4 - Teitz, Guillet & ModeloExato\arquivo',
'Visitas');
ENDDATA

!Minimiza a distancia total percorrida;
[Fo] MIN=@SUM(matriz(i,j):Visita(i)*dist(i,j)*x(i,j));

!Restricoes;
@FOR(Cluster(j):@SUM(matriz(i,j):x(i,j))>=CapCluster(j));
@FOR(Cluster(j):@SUM(matriz(i,j):x(i,j))<=33);
@FOR(cidades(i):@SUM(matriz(i,j):x(i,j))=1);

@SUM(matriz(i,j):x(i,j))=@SIZE(Cidades);

@FOR(matriz(i,j):@BIN(x(i,j)));

DATA:

! Exporta dados para o Excel;
@OLE('E:\Metodologia 4 - Teitz, Guillet & ModeloExato\arquivo.xls',
'Solucao') = x;
ENDDATA
END

Anexo 6: Implementação Computacional em LINGO do Critério Superior

MODEL:

TITLE: Criterio Superior;

SETS:

Cidades / @OLE('C:\DadosFinais\Metodologia 1 Limitar
Superiormemente\BaseDados1.xls', 'cidades') /:Tlinha,Tcoluna;
Veic / @OLE('C:\DadosFinais\Metodologia 1 Limitar
Superiormemente\BaseDados1.xls', 'Veic')/;;
matriz(Cidades,Cidades): dist,x,Resultado,Solucao;
ENDSETS

DATA:

!Importa dados do Excel;

dist,T,Tlinha,Tcoluna= @OLE('C:\DadosFinais\Metodologia 1 Limitar
Superiormemente\BaseDados1.xls', 'Solucao', 'Veic','Tlinha','Tcoluna');
ENDDATA

! Minimiza a distancia total percorrida;

[Fo] MIN=@SUM(matriz(i,j)|dist(i,j)#ne#0:dist(i,j)*x(i,j));

! Restricoes;

@FOR(matriz(i,j)|dist(i,j)#EQ#0:x(i,j)=0);

@FOR(Cidades(j): @SUM(Cidades(i): x(i,j))= Tcoluna(j));

@FOR(Cidades(i): @SUM(Cidades(j): x(i,j))=Tlinha(i));

! A variavel x(i,j) deve ser binaria;

@FOR(matriz(i,j):@BIN(x(i,j)));

@FOR(matriz(i,j):Resultado(i,j)=x(i,j)*dist(i,j));

DATA:

! Exporta dados para o Excel;

@OLE('C:\DadosFinais\Metodologia 1 Limitar Superiormemente\BaseDados1.xls',
'dist') = resultado;
ENDDATA

END

Anexo 7: Implementação Computacional em LINGO do Modelo PL → PLI

MODEL:

TITLE: Modelo PL PLI;

SETS:

idades / @OLE('C:\DadosFinais\Metodologia 3 Critério PL
PLI\BaseDados1.xls', 'idades') /:DemandaPL;

VeicPL / @OLE('C:\DadosFinais\Metodologia 3 Critério PL
PLI\BaseDados1.xls', 'VeicPL')/;

matriz(idades, idades): dist,x,f, Parametro, Solucao;

ENDSETS

DATA:

!Importa dados do Excel;

dist,CapVeicPL,DemandaPL, TPL= @OLE('C:\DadosFinais\Metodologia 3 Critério
PL PLI\BaseDados1.xls','dist', 'CapVeicPL','DemandaPL','VeicPL');

ENDDATA

! Minimiza a distancia total percorrida;

[Fo] MIN=@SUM(matriz(i,j)|dist(i,j)#ne#0:dist(i,j)*x(i,j));

!Restrições;

@FOR(matriz(i,j)|dist(i,j)#EQ#0:x(i,j)=0);

@FOR(idades(i): @SUM(idades(j): x(i,j)) = DemandaPL(i));

@FOR(idades(j): @SUM(idades(i): x(j,i)) = DemandaPL(j));

@FOR(idades(j): @SUM(idades(i): f(i,j)) - @SUM(idades(i): f(j,i)) =
demandaPL(j));

@FOR(matriz(i,j): f(i,j) = CapVeicPL*x(i,j));

@SUM(matriz(i,j): x(i,j))=CapVeicPL;

DATA:

!Exporta dados para o Excel;

@OLE('C:\DadosFinais\Metodologia 3 Critério PL PLI\BaseDados1.xls',
'Parametro') = x;

ENDDATA

END

Anexo 8: Implementação Computacional em Matlab®7 da Análise de *Clusters*

```
% Obter Dados
Dados=xlsread('Floyd','B:HC');
K=input('Quantos Grupos?');
n=input('Quantas Cidades?');

%Separa em grupos de acordo com a estatística:
Sd=input('Qual é a Sede:');
for i=1:n
    Dados(i,i)=0;
end
Distancia=squareform(Dados);
Z1=linkage(Distancia,'single');
Z2=linkage(Distancia,'complete');
Z3=linkage(Distancia,'average');
Z4=linkage(Distancia,'centroid');
Z5=linkage(Distancia,'ward');

Cluster1=cluster(Z1,K);
Cluster2=cluster(Z2,K);
Cluster3=cluster(Z3,K);
Cluster4=cluster(Z4,K);
Cluster5=cluster(Z5,K);
clc
LigacaoSimples=cophenet(Z1,Distancia)
LigacaoCompleta=cophenet(Z2,Distancia)
LigacaoMedias=cophenet(Z3,Distancia)
LigacaoCentroid=cophenet(Z4,Distancia)
LigacaoWard=cophenet(Z5,Distancia)

% Gera a matriz para o modelo exato:
Cluster2=[zeros(n,1),Cluster5];
for i=1:n
    Cluster2(i,1)=i;
end
for i=1:n
    for j=1:n
        if Cluster2(i,2)~=Cluster2(j,2) & i~=Sd & j~=Sd
            Dados(Cluster2(i,1),Cluster2(j,1))=0;
            Dados(Cluster2(j,1),Cluster2(i,1))=0;
        end
    end
end
for i=1:n
    Grupos(i,Cluster2(i,2))=i;
end
Grupos=-sort(-sort(Grupos));
d=size(Grupos);
teste1=[];
teste2=[];
for i=1:d(1)
    if max(Grupos(i,:))==0
        teste1=[teste1;i];
    end
end
for i=1:d(2)
    if max(Grupos(:,i))==0
        teste2=[i;teste2];
    end
end
```

```
end  
Grupos(teste1,:)=[];  
Grupos(:,teste2)=[];
```

Anexo 9: Implementação Computacional em LINGO do Modelo Exato de Roteamento

MODEL:

TITLE: Roteamento de Veiculos;

SETS:

```

cidades / @OLE('C:\DadosFinais\Metodologia 3 Critério PL
PLI\BaseDados1.xls', 'cidades') /:DemandaPI;
Veic / @OLE('C:\DadosFinais\Metodologia 3 Critério PL PLI\BaseDados1.xls',
'VeicPI')/;;
matriz(cidades, cidades): dist,x,f;
ENDSETS

```

DATA:

```

!Importa dados do Excel;
dist,CapVeicPI,DemandaPI, TPI= @OLE('C:\DadosFinais\Metodologia 3 Critério
PL PLI\BaseDados1.xls', 'Resposta', 'CapVeicPI', 'DemandaPI', 'VeicPI');
ENDDATA

```

! Minimiza a distancia total percorrida;

```
[Fo] MIN=@SUM(matriz(i,j)|dist(i,j)#ne#0:dist(i,j)*x(i,j));
```

!restricoes que viabilizam a resolucao do problema pelo modelo exato;

```
@FOR(matriz(i,j)|dist(i,j)#EQ#0:x(i,j)=0);
```

! De uma cidade i, exceto o deposito, so sai um unico veiculo;

```
@FOR(cidades(i) | i #ne# 1: @SUM(cidades(j)|dist(i,j)#ne#0: x(i,j)) =
DemandaPI(i));
```

! A uma cidade j, exceto o deposito, so chega um unico veiculo;

```
@FOR(cidades(j) | j #ne# 1: @SUM(cidades(i)|dist(i,j)#ne#0: x(j,i)) =
DemandaPI(j));
```

! O deposito deve ter uma frota de TPI veiculos;

```
@FOR(cidades(j) | j #EQ# 1: @SUM(cidades(i)|dist(i,1)#ne#0: x(i,j)) = TPI);
@FOR(cidades(i) | i #EQ# 1: @SUM(cidades(j)|dist(1,j)#ne#0: x(i,j)) = TPI);
```

!O numero de veiculos que saem do deposito deve ser igual ao numero de veiculos que chegam ao deposito;

```
@SUM(cidades(j)|dist(1,j)#ne#0: x(1, j)) = @SUM(cidades(j)|dist(j,1)#ne#0:
x(j, 1));
```

! Ao passar por uma cidade j, exceto o deposito, o veiculo deve atender a demanda dessa cidade, isto e, deve deixar demanda(j) unidades de produto na cidade j;

```
@FOR(cidades(j) | j #ne# 1: @SUM(cidades(i)|dist(i,j)#ne#0: f(i,j)) -
@SUM(cidades(i)|dist(j,i)#ne#0: f(j,i)) = DemandaPI(j));
```

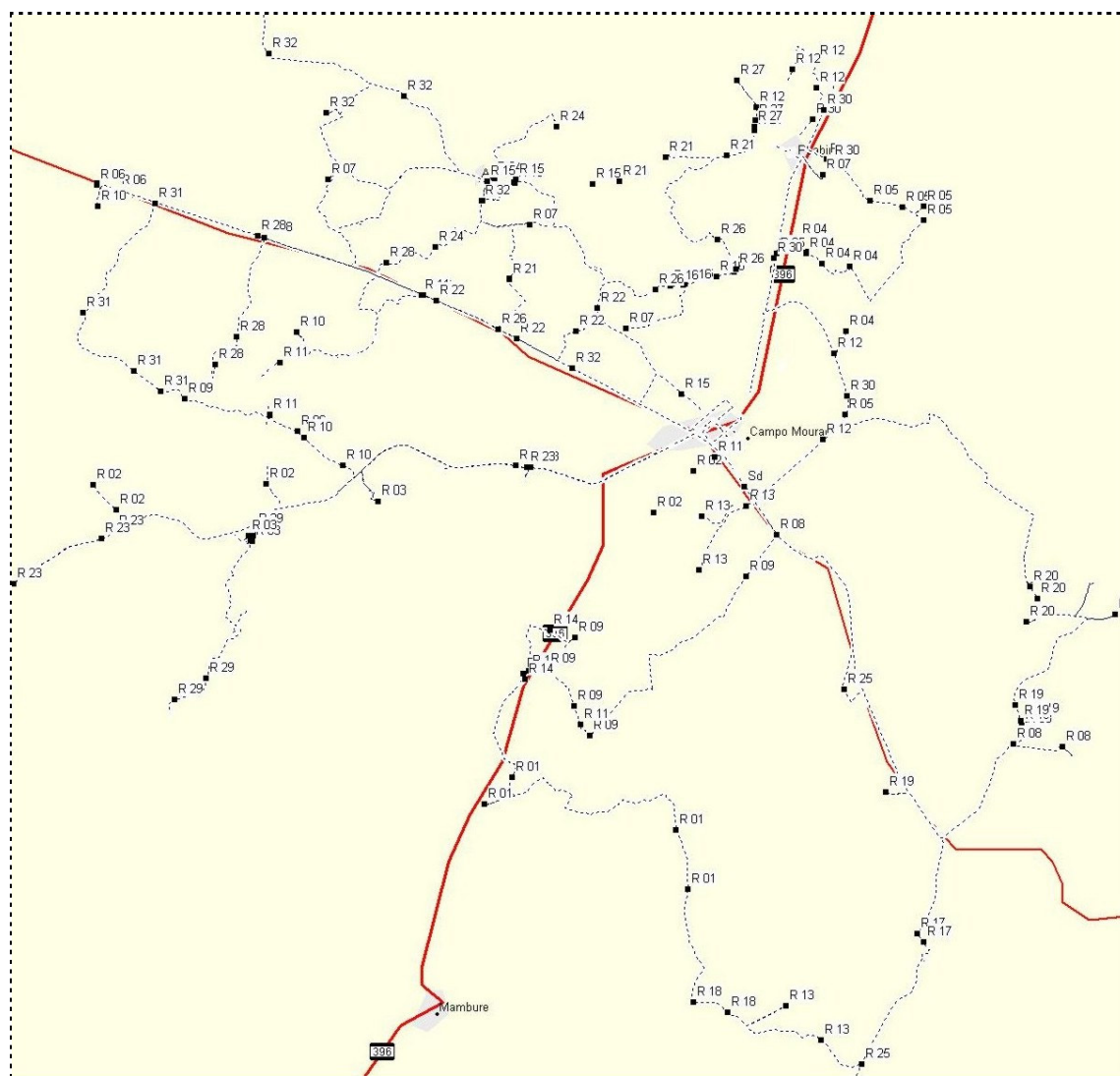
! A quantidade de fluxo de i para j nao pode superar a capacidade do veiculo;

```
@FOR(matriz(i,j)|dist(i,j)#ne#0: f(i,j) <= CapVeicPI*x(i,j));
```

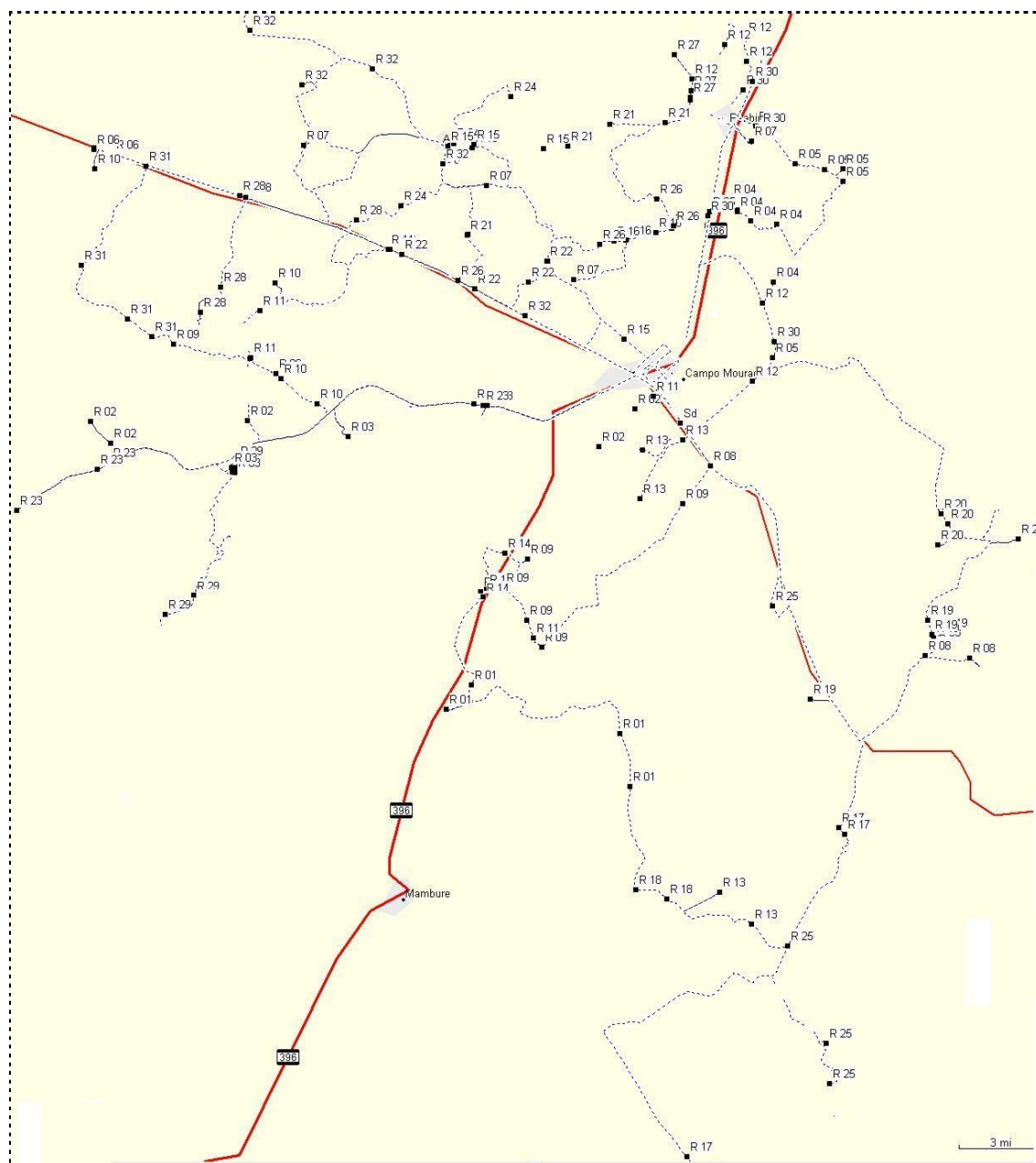
! A variavel x(i,j) deve ser binaria;

```
@FOR(matriz(i,j)|dist(i,j)#ne#0:@BIN(x(i,j)));
```

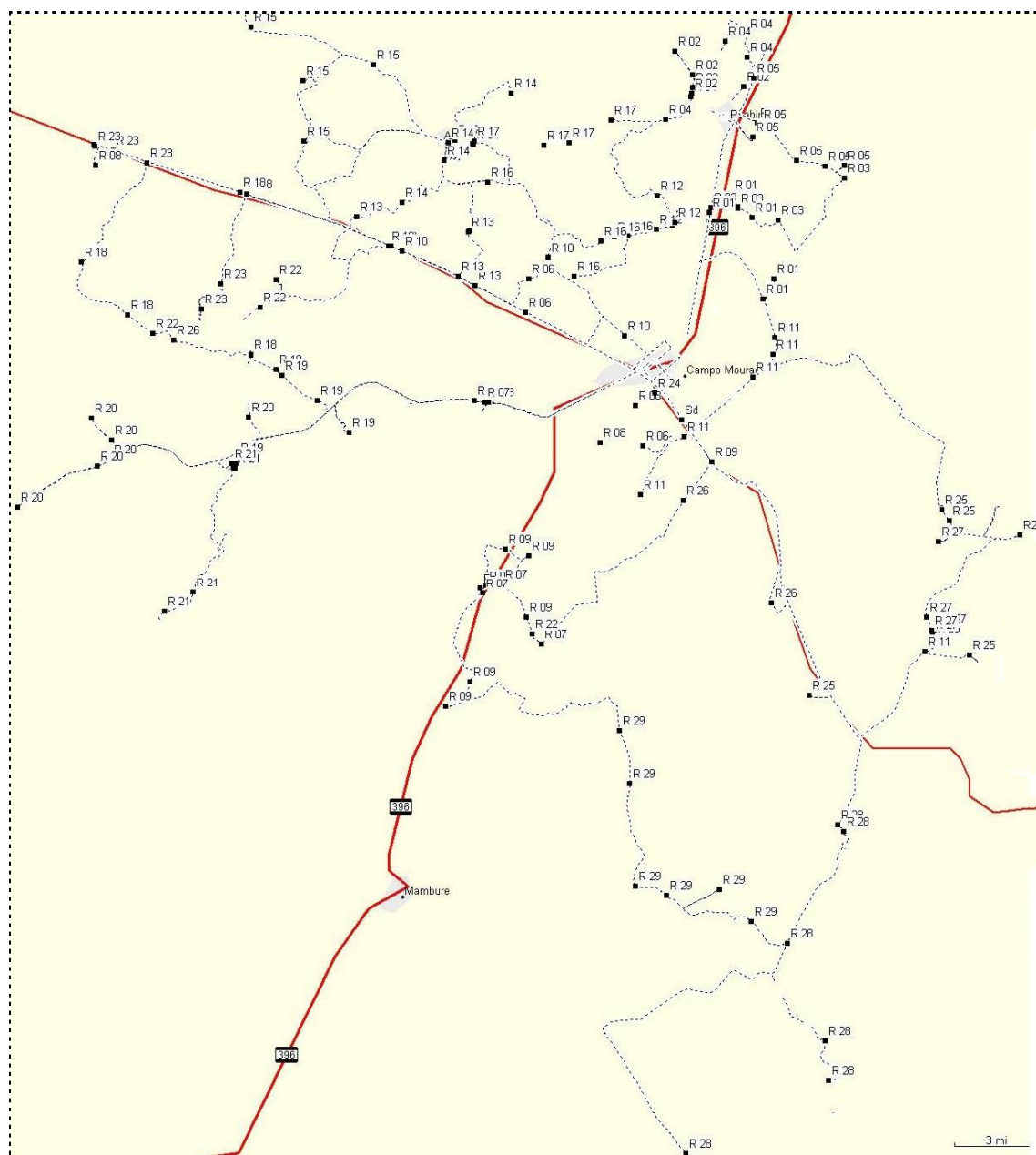
END

Anexo 10: Visualização gráfica das rotas indicadas na tabela 5.2

Anexo 11: Visualização gráfica das rotas indicadas na tabela 5.4



Anexo 13: Visualização gráfica das rotas indicadas na tabela 5.8



Anexo 14: Visualização gráfica das rotas indicadas na tabela 5.10

